

ROLLING RHINO The Ubuntu spin that's always updated



GREEN TUX How to buy a greener Linux box



BACK TO BASIC We chart the rise and fall of the classic code





STAY SAFE ONLINE

Eat adverts, trackers and snoopers using Pi-hole for homewide network protection



PLUS: HOW TO

- >> Build your own musical mini Pi-ano
- >> Expand your LucasArts-style adventure
- >> Add redirection to the LXF terminal

HOME MOTION CCTV

Design, build and control the ultimate home CCTV

TORRENT BOX

Remote-manage your downloads

CROSS-PLATFORM

Develop apps that will run on everything!





We're working to create a future where brain tumours are curable. We urgently need your help to accelerate research.

Text DEFEAT5 to 70507 to donate £5, please help us to find a cure.

thebraintumourcharity.org







MEET THE TEAM

We're all about taking back control this issue, implementing Pi-hole protection. What open source tools do you use to help bring control back to your life?



Matthew Holder

On my ancient HP MicroServer, I run FreeNAS, so I can centrally store my data. RClone enables me to copy data to and from various cloud platforms. Completing the package is *Home Assistant* and various

containers, so I can tie all of my smart home gear together.



Les Pounder

I use Pi-hole on an old Raspberry Pi behind the living room TV. Why? Well, I'm sick of adverts popping up everywhere, and running ad-blocking software on every device is a chore. Install Pi-hole,

point the devices to it, and bang! The ads are gone!



Michael Reed

I get a lot of my information from books, and I have a huge collection. Much of the library I've amassed is out of date by most people's standards, but I like the idea of learning about things from the mindsets of

different eras to gain a unique perspective.



David Rutland

I run open source Pi-hosted software for almost everything. Trilium is my latest find, and I use Snikket for family comms; Immich as a better Google Photos; Snappymail/ Postfix/Dovecot for email; Jellyfin and

Audiobookshelf; FreshRSS; Nextcloud - and so much more.



Nick Peers

My Ubuntu Server-powered J5040-ITX sits at the heart of my digital life, and the key thing giving me control over it (and most of the services it runs, from Nginx Proxy Manager to Bitwarden) is Docker - soon to

be Podman. Containers make rolling out new services easy.

Corporate takeover



Who said history repeats itself? First there was Netscape and everyone used it and it turned out bad. Then there was Internet Explorer and everyone used it and it turned out bad. Now everyone's using Chrome (laughs in Firefox) and Google is about to effectively ban ad blocking, which is bad. You can read more about the Manifest v3 plans and more in our news, but beyond using Firefox for

general browsing, we're looking at how you can protect your entire network from the corporate takeover with Pi-hole.

This gives you better control over what can enter your network and it blocks malware at the gate before it can even get to any devices. So, you have less to worry about when you have family members using phones, tablets, computers, smart TVs and more. Never mind the options to plug in VPNs and browsing controls to protect your little ones, too.

But, more widely, we're all about handing control back to you, the user, from controlling your own home CCTV system to controlling your system updates with Rhino Linux, and your torrent downloads. Or why not take control of your own pointand-click adventure? There are almost endless possibilities with open source and Linux, so no matter what 2024 holds, there's plenty for us to enjoy.





Subscribe & save! On digital and print:

see page 16

www.techradar.com/pro/linux February 2024 LXF311 | 3

Contents



REVIEWS



Intel Core i5 14600K 19

No one calls **Jacob Ridley** mid-range! And this mid-range chip has a fight on its hands if it wants to justify its existence.

EndeavourOS 11-2023

Nate Drake endeavours to discover all that is great about this Arch-based Linux distro,

SysLinuxOS 12.2 21

created by former Antergos developers.

Nate Drake dives into SysLinuxOS to find whether its reputation as the dream OS for system administrators is well deserved.

OpenMandriva Lx 5.0 22

Nate Drake relives Mandriva Linux's heyday with its programmer and community-developed successor.

Ultramarine Linux 39 23

Nate Drake decides to take this Thai Fedora-based distro for a spin and finds it fast, simple and surprisingly customisable.

Steam Deck OLED 24

Management is overjoyed: "Take an old product and slap a new badge on for more money?" **Tyler Colp** likes his job too much to correct them.

ROUNDUP



File managers 26 Michael Reed has written widely about Linux and open source software for many years, and his files are always well managed.

STAY SAFE ONLINE!

Block unwanted online attention by sending its packets to the Pi-hole. **David Rutland** helps you protect your browsing data with the ultimate paranoid's delight! See page 32!



IN DEPTH



Rise and fall (and rise) of Basic 50 We plot the symbolic milestones of the

most basic of all-purpose code with the informative instruction of the multi-lingual **Mike Bedford**.



Roll out the Rhino!

In an increasingly crowded marketplace, **Nick Peers** discovers why this new Ubuntu derivative could turn out to be a barrel of fun! CREDIT: Getty Images/ Brand X Pictures/Rubberball. Dartmouth College. Rhino Linux. Magictorch



Ubuntu 23.10 for Pi ______ 42

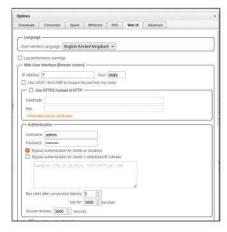
The dapper **Les Pounder** has been using Ubuntu since Dapper Drake and has been looking forward to this Pi-based iteration.

Bambu Lab A1 Mini43

Good things come in small packages is something **Denise Bertacchi** never says – now she's eating her delicious words.

Build a Pi Pico electronic piano 44

Les Pounder is trying his best to make his mum proud by not taking apart an electronic piano!



Torrent your slice of Pi46

Torrenting isn't all about taking – **Christian Cawley** helps you use a Pi to serve open source ISOs on the BitTorrent network.

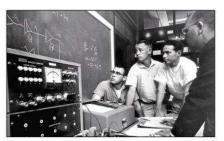
o electronic piano 44

TUTORIALS

TERMINAL: Total shell recall 54

Too busy shouting, "Get your ass to Mars," **Shashank Sharma** thinks a strong shell history can be a real life saver.

Nick Peers reveals how to bypass expensive (and insecure) cloud security systems to create your own.



EMULATION: Analogue computer

Today, computers are nearly always digital, but analogue computers had their place, as **Mike Bedford** discovers via emulation.

ADVENTURE GAME: Inventories 66

Nate Drake invites you to relive the glory days of point-and-click adventure games by creating your very own.

CODING ACADEMY

Write your own Android application 90 Matt Holder investigates how to create

your very own Android application using Python and the Kivy framework.

It's all redirection! 94

Not afraid of the depth of Linux systems, Ferenc Deak continues his journey to show how to implement redirection in the shell.



REGULARS AT A GLANCE

News 6

Google blocks ad blockers, X.Org set to be ex-Org, *GIMP 3* inches towards release, PipeWire 1.0 is no pipe dream, plus industry insiders' opinions and a look at some of the latest distro releases.

Answers 11

Neil Bothwick is considering changing his name to Tux, but such nonsense doesn't stop him from answering queries about database design, data destruction, digits, differences and more...

Mailserver14

Readers continue to talk languages, plus ask advice about scanners and Linux.

Subscriptions 16 Grab your monthly dose of Linux and save

Grab your monthly dose of Linux and save 50% off the usual price!

Back issues 64 Get hold of previous *Linux Format* editions.

Mayank Sharma can't go out and risk being choked by the toxic air, so instead makes do by finding apps that are a breath of fresh air: Recoll, Polybar, Curtail, Betterbird, BleachBit,

FileZilla, JDownloader 2, Gods Deluxe, Naval

Next month 98

Battle, Fantascene and Audacity.

ADMINISTERIA

Administeria74

Stuart Burns explains why he's ditched his MacBook Pro in favour of a second-hand ThinkPad, plus offers advice for sysadmins.

AccuWeb Hosting76

Offering power-packed web hosting for all levels of user, including the ever-sopicky **Mike Williams**.

Smartproxy......77

Smartypants **Sead Fadilpašić** tests a versatile and fairly affordable proxy service.

Green PC machines 78

The world's biggest PC maker says it can achieve net zero by 2050. **Barry Collins** visits its US headquarters.

Newsdesk

THIS ISSUE: Google blocks ad blockers >> X.Org set to be ex-Org >> GIMP 3 inches towards release >> PipeWire 1.0 is no pipe dream

EXTENSIONS

Google steps up arms race against ad blockers

First ad blocker users experienced interrupts, then YouTube videos loaded slower, now Google is disabling most ad blockers altogether.

anifest 3 is the latest version of *Chrome's* extensions platform. The Chrome for Developers' website describes it as an improvement to the "privacy, security and performance of extensions". This claim has been met with scepticism by the EFF and other privacy advocates. The controversy centres around Manifest 3's removal of an extension's ability to use remotely hosted code.

As of *Chrome 127*, Google has also given itself the right to review and authorise browser extension updates. This gives it the power to slow down or disable ad-blocking extensions.

The EFF describes the update as a "raw deal" for users and another example of the "inherent conflict of interest that comes from Google controlling both the dominant browser and one of the largest internet advertising networks".

This isn't the first time Google has inadvertently endangered privacy in the name of protecting it. Its Privacy Sandbox was designed to create a more private web by replacing third-party cookies but it can, in fact, make it easier for advertisers to target individual users.

There's little evidence to suggest that limiting ad-blocking extensions will improve browser performance. In its rebuttal, the EFF cites a 2020 Princeton/University of Chicago study that shows privacy extensions can actually improve browser performance.

Speaking on the AdGuard company blog, Andrey Meshkov stated: "Nearly all browser extensions as you know them today will be affected in some way: the more lucky ones will 'only' experience problems, some will get crippled, and some will literally cease to exist."

This is particularly bad news for YouTube users, who until now have been able to avoid paying \$13.99 per month for its ad-free Premium version.

One alternative would be to switch browser, but given how many others, such as *Microsoft Edge*, are based on *Chromium*, this won't be easy. The developers of privacy-focused *Brave* have pointed out their Shields feature is built into the browser, so doesn't need to use the extension API.



Manifest 3 will launch in June 2024 on Google Chrome. Other Chromium-based browsers, such as Edge, will also deploy it.

DOWN THE TUBE

This is bad news for YouTube users, who until now have been able to avoid paying for its ad-free Premium version.

Mozilla has also confirmed that it'll be implementing its own version of Manifest 3 to ensure cross-browser compatibility, as well as continuing support for Manifest 2. It has also confirmed that privacy extensions like uBlock Origin will continue to function.

Manifest 3 also won't affect DNS-level ad blocking, such as that used by *Pi-hole*. By coincidence, we cover how to set up your free ad-blocking DNS server via *Pi-hole* this issue.



DNS-based ad blocking doesn't require extensions, so solutions like Pi-hole will be unaffected (see page 32 for more).

6 | **LXF311** February 2024 www.linuxformat.com

WINDOWING SYSTEM

X.Org being phased out

For four decades, the X Window System has endured despite competition, but major OSes are bowing out.

n late November, Red Hat engineer Carlos Soriano Sanchez posted on the company's blog to confirm that as of RHEL 10, X.Org server and other X servers (except Xwayland) will be removed.

Sanchez also uploaded a number of posts to his Mastodon channel detailing the work involved in maintaining the X11 infrastructure, which requires a full-time dedicated engineer. He justified Red Hat's decision because of the extra burden on the quality engineering team, which currently has to test both Wayland and X.Org Server.

This news was hardly surprising, given that Wayland has been used by default in most cases of RHEL 8 and that X.Org Server was officially deprecated as of RHEL 9.

Still, it marks a wider move away from X11. In November, developer Nate Graham also confirmed on the KDE blog that the new Plasma 6 desktop will also default to Wayland sessions, with preliminary support for playing HDR-capable games on compatible screens. Naturally, this will affect any OS that deploys Plasma 6, such as Fedora 40. Graham used the blog comments section to clarify that X11 sessions

will be available in the near future, however support will definitely be "thrown away at some point in the next 10 years".

Merge requests and developer conversations suggest that Gnome also seems to be taking steps towards dropping X11. Since 2016, Gnome sessions have defaulted to Wayland. Developers also argue that X11 has received less testing in recent years and that removing support would reduce the desktop's code by several thousand lines.

X.Org continues to be actively developed. For the latest news, visit https://lists.x.org/mailman/ listinfo/xorg-devel.



As of v10, Red Hat Enterprise Linux won't support X11. Other major OSes are phasing it out in favour of Wayland.

SOFTWARE)

Road map to GIMP 3

Latest version of GIMP finally has a release schedule.

n 21st November, Jehan Pages, the lead developer of the *GNU Image Manipulation Program (GIMP)*, announced on Patreon that the muchanticipated *GIMP* 3.0 is tentatively scheduled for release at the Libre Graphics Conference in May 2024.

In broad terms, the plan involves releasing a development version in December 2023. Naturally, this will require a GUI feature freeze, which is set for mid-December, allowing coders to halt new feature development and focus on bug fixes. However, this means that long-awaited features, such as full support for non-destructive filters attached to layers, may have to be included in a patch release instead of being part of *GIMP 3.0* itself.

GIMP's new API allows it to make much better use of colour spaces and models. However, rewriting the API will allow little time for bug fixes. This may be why the feature freeze will not include API changes. Pages has stated this is because "an additional month for the last little tweaks may be worth it".

These "tweaks" may not be so little, though. At the time of writing, *GIMP*'s GitHub page records over 4,000 open issues, an increase of around 30% in the past year. Around 600 of these can cause the application to crash, all of which would presumably have to be resolved in a new, stable release.

View the full GIMP 3.0 development timeline as well as the latest news at https://gitlab.gnome.org/GNOME/gimp/-/issues/10373.

OPINION

ALL AI HYPE



Joe Brockmeier is head of community, Percona.

My crystal ball is only slightly better than a second-hand Magic 8 Ball, but gazing into it for a preview of 2024 provides a little clarity. Namely, non-stop hype for Al/ML and direct impacts on open source.

Money and resources follow hype, so projects and efforts that don't have an AI/ML cast are going to have a harder time finding support. This is worse than previous hype cycles around things such as Linux containers, Kubernetes and open source laaS – which all solve real and immediate problems.

AI/ML, on the other hand, is mostly a solution in search of a problem – or a solution to the problem of 'vendors want more revenue' – while there's plenty of real problems needing attention.

Case in point, the real need for more maintainers, participation and testing of projects that we're all using in production today. A data loss bug made it into stable Linux kernels and all the way to a Debian release. Mistakes happen, but one wonders whether if some of the resources being firehosed on to Al/ML tech had been focused on crucial but boring maintenance and testing, it could have been prevented.

The tension that's brewing between commercial interests and community health is going to be a major focus in 2024, and I don't need ChatGPT to tell me that.

OPINION

ALL IN THE PIPE!



George Kiagiadakis is a principal software engineer at Collabora.

PipeWire 1.0 was released late November (see right), a major milestone in a long journey that I had the opportunity to be a part of. When rumours about PipeWire being a possible successor to PulseAudio had begun circulating in 2018, I was working on a project that was using PulseAudio in an odd way and wanted to explore other solutions. The first PipeWire hackfest was announced that summer and I signed up for it without hesitation. The rest is history.

Thanks to a strong collaboration with Automotive Grade Linux (AGL), the WirePlumber session manager was born shortly after. PipeWire has strong potential to solve problems outside the scope of the Linux desktop, and WirePlumber is a key component in that. By being the orchestrator behind PipeWire's graph, it was designed to be able to customise the behaviour of the entire system so that it can address different use cases, such as the ones in the automotive realm.

The 1.0 release now marks a point where we can confidently say that PipeWire is here to stay. But it's not the end of the journey. There are many new areas to explore going forward, especially in WirePlumber and the ecosystem that builds around PipeWire.

MEDIA

PipeWire turns 1.0

The audio/video bus designed to replace PulseAudio, JACK and other systems has now reached version 1.0.

n 2015, Wim Taymans of Red Hat began work on PipeWire, a server designed to handle audio/video streams on Linux. It was envisaged as a successor to PulseAudio, which had experienced issues with high CPU usage and Bluetooth connectivity.

Since then, PipeWire has proved a reliable choice for Linux desktops. On 26th November, the release of version 1.0 was announced on the project page of **Freedesktop.org**. Version 1.0 is API and ABI compatible with previous 0.3.x releases.



PipeWire 1.0's new features include an option for exposing ALSA controls as prop parameters.

In an interview with Fedora Magazine,
Taymans confirmed that version 1.0 also aims
to "close the gap" between PulseAudio and
PipeWire. This work included adding modules
for features such as echo-cancellation, S/PDIF
pass-through, AirPlay support and multiple
sample rates. He claims that most of these new
modules now have more features than the
PulseAudio equivalents.

Other improvements include Bluetooth LC3 codec and compatibility improvements, improved transport and time handling, as well as buffer reuse for JACK.

HARDWARE

More Linuxfirst laptops

Linux come preinstalled on more and more machines.

uxedo Computers belongs to a small but growing niche of hardware vendors that specialise in selling new hardware with Linux preinstalled. Currently, the company offers both Ubuntu and its own Tuxedo OS, which uses Ubuntu with the KDE Plasma desktop.

The Sirius 16 Gen-5 comes with a 16.1-inch display (2,560x1,440), an 80Wh battery and up to 96GB of DDR5-5600 RAM, as well as up to 8TB of PCle 4.0 SSD storage. It also incorporates USB 4.0, HDMI 2.1 and Wi-Fi 6E.

Learn more at www.tuxedocomputers.com.



The Sirius 16 is the first all-AMD Linux gaming laptop with Ryzen 7 7840HS CPU and Radeon RX 7600M XT graphics.

FIRMWARE

Open firmware doing very well

100 million firmware updates supplied by the LVFS.

n 2015, Richard Hughes created the LVFS (Linux Vendor Firmware Service) and the fwupd tool to easily update Linux firmware. The service is used by a number of major distros to provide metadata for clients such as fwupdmgr and Gnome Software.

It also doesn't charge vendors for hosting content. Richard has stated in the Gnome blog that two or three vendors sign up every month.

Although the LVFS has supplied updates to 100 million Linux machines worldwide, the real number could be much greater given that users can redistribute updates without any tracking.



The LVFS website has a database that you can use to check if there's compatible firmware for your chosen hardware.

CREDIT: Tuxedo Computers

Distro watch

What's behind the free software sofa?

KALI 2023.4

Beloved by pen-testers everywhere, Kali's latest iteration includes Gnome 45, giving it the benefit of better search features in *Nautilus* and full-height sidebars. Kali Linux AMD64 and ARM64 will also now be available on Amazon AWS and Microsoft Azure marketplaces. There's a Raspberry Pi 5 image, too. Kali also now uses mirrorbits to optimise installation and updates of packages. There are a number of new security tools as well. Learn more at www.kali.org.



Kali is a Debian-based distro with security and forensics tools.

MABOX 23.12

Mabox is a relatively new operating system, having first been released as recently as 2020. It's designed to be a lightweight Linux distribution, featuring a customised desktop based on *Openbox*. The latest version uses the LTS version of the Linux kernel (6.6) but there's also an ISO image available for older hardware that uses kernel 5.4. Mabox utilises Tint2 and Jgmenu to create powerful customisable panels, upgraded *Conky* widgets and a *Quake*-style console. You can find out more at https://maboxlinux.org.



Mabox is a Manjarobased distribution that uses Openbox.

ARMBIAN 23.11

Armbian is designed for a wide variety of popular ARM-based devices, including Banana Pi, CubieBoard, Olimex, Orange Pi, Odroid, Pine64 and others. The latest version, code-named Topi, expands this further to include the NanoPi R6S/R6C, TI SK-TDA4VM and Xiaomi-elish, among others. Daily image builds are also now available based on Ubuntu Mantic or Debian Trixie, and the OS now features fixed display managers across all desktops. See www.armbian.com to learn more.



Based on Debian 12, Armbian is designed for ARM boards.

SPARKYLINUX 7.2

The team at SparkyLinux has been releasing stable versions of its lightweight operating system for around five years. It uses customised Enlightenment and LXDE desktops. The latest version, code-named Orion Belt, is a quarterly updated point release of Sparky 7, based on Debian 12 Bookworm. PC versions now use version 6.1.55 of the Linux kernel. The distribution also now sports *LibreOffice 7.4.7*, and the *Calamares* installer has been upgraded to v3.2.61. You can read more at https://sparkylinux.org.



SparkyLinux is based on Debian and has a variety of desktops.

ZORIN OS 17 BETA

Zorin is popular with Linux newcomers as it has a Windows-like GUI. It can even run Windows programs, as it supports Wine and *PlayOnLinux*. The latest community edition includes support for universal search including files, apps, contacts and appointments. Workspaces are now arranged horizontally, and you can open the overview and use gestures to switch between them. The new spatial desktop allows switching between workspaces in 3D, too. Learn more at: https://zorin.com/os.



Zorin is based on Ubuntu and designed for Linux noobs.

OPINION

NEW YEAR! NEW LIBRE!



Italo Vignoli
is one of the founders
of LibreOffice and the

Like every year, there will be two new major releases of *LibreOffice* in 2024 – version 24.2 in February and version 24.8 in August– the numbering no longer being progressive.

In 2024, there will be many accessibility-related innovations, the result of the Document Foundation's investment in this area, with a dedicated developer. Accessibility is a fundamental factor for software like *LibreOffice*, which aims to be a reference for everyone, especially those who need support because they are members of a minority.

Of course, there will continue to be innovations related to interoperability with *Microsoft Office's* proprietary formats, which continue to be a problem due to their poor or non-existent adherence to the standard. *LibreOffice* offers an excellent level of compatibility, the result of constant analysis and benchmarking by the devs.

Finally, there will be more communication activity on security, because we have noticed that the majority of users, starting with corporate users, are not sufficiently informed and therefore tend to underestimate the work that is done on a daily basis on this fundamental aspect of the suite.

OPINION UPSTREAM FLINK



Jon Masters is a kernel hacker who's been involved with Linux for over 22 years.

A recent data corrupting bug in the ext4 filesystem served as a timely reminder about the benefits and limitations of 'stable' kernels. The goal of the 'stable' kernels is to allow end users and developers to run with the latest and greatest, without having to track upstream.

The problem turned out to be an innocent-looking patch that had been added in 6.1.64, but that had a subtle dependency upon another kernel patch that had been omitted in the backporting process. Consequently, Debian users (most worryingly, those who opt for unattended automatic deployment) began to see corruption on their ext4 filesystems with a potentially quite significant impact. The community had some pushback about the lack of good upstream focused automated testing that might have caught this.

Contrast this with Enterprise Linux kernels, which are often criticised for being necrokernels (and other less printable terms), but that exist because a vendor such as Red Hat or SuSE stands behind them with automated testing validating each release. It's a good opportunity to remind ourselves that vendor kernels are not all bad, and occasionally their value is reinforced. Until upstream testing changes, there's real benefit in Enterprise Linux for critical stuff.

Kernel Watch

Jon Masters keeps up with all the latest happenings in the Linux kernel, so you don't have to.

inus Torvalds announced one of the final Release Candidates for what will be Linux 6.7 (-rc5), making us on track to start the new year with a shiny new 6.7 kernel, and the corresponding two-week merge window for new features that will eventually stabilise into 6.8.

Just in time for the new year, the Linux Kernel Mailing List (LKML) is moving servers as **vger.kernel.org** is finally decommissioned. Hopefully folks won't notice the transition to modern infrastructure.

That was 2023

2023 was another very busy year in the Linux community. It started with the Linux 6.2 kernel cycle, the first to include baseline useful bits of Rust language support.

Also included was Apple Silicon M2 Macs (6.4, in May) and mount-beneath (6.5, in September) for updating the underlay layers of overlay filesystems used to construct containers – in other words, allowing for the update of container base layers more easily. The year rounded out with Linux 6.7, which includes the removal of Itanium architecture support and the addition of a new filesystem called bcachefs, covered last month.

There was a number of broad themes in developments throughout the year, among them growing momentum for Rust as an alternative for certain Linux driver code, upstream support for various Confidential Computing (CoCo) implementations from

the various architectures (Intel TDX, AMD SEV-SNP, Arm's Realms and RISC-V's CoVE), and the continuation of CPU security vulnerabilities as well as their mitigations. On the latter front, we witnessed yet another vulnerability just this past month in the form of a SLAM (Spectre based on Linear Address Masking), which attacks a feature only recently merged into the kernel for certain upcoming CPUs.

Looking on to the year ahead, it seems likely there will be growing debate around upcoming European legislation named CRA (Cyber Resilience Act) since it has implications in terms of potential liability for individual open source developers who contribute to large projects. Like many such laws and regulations, the CRA is well meaning, aiming to improve the overall robustness of software (within the EU), but it might just have a number of unintended consequences and fallout as it is implemented.

It seems likely 2024 will see growing adoption of Rust across the industry, and in particular within Linux. The language is even beginning to have a true formal specification and interest among safety critical users, such as in automotive. There was even a story recently about a variant of real-time Linux kernel support leveraging a hard real-time hypervisor written in Rust going to space. Finally, the year ahead will likely continue to see fallout from the various 'efficiency' drives happening across industry, and a pivot to AI all the things, which hopefully won't have too much negative impact upon open source investment.

» ONGOING DEVELOPMENT

It appears we are getting closer to the removal of the venerable SLAB. As Vlastimil Babka wrote back in November, "The SLAB allocator has been deprecated since 6.5 and nobody has objected so far." SLAB was originally inspired by the Solaris allocator of the same name. It creates small in-memory caches of similar sized kernel objects such that they can be initialised once and then rapidly recycled without paying the initialisation tax repeatedly.

Kanchan Joshi posted updated patches implementing "copy offload support" for block devices such as NVMe drives where

the device itself can accelerate the process of copying existing data stored within the media. The patches include support for emulation of the feature in cases "when offload is natively absent".

Philipp Stanner posted "Regather scattered PCI-Code" to clean up the PCI subsystem, moving various functions into generic places and fixing fragmentation of code that lives partly in devres (device resource) but is only used by the PCI subsystem. Philipp says, "I hope this is OK. If we can get it in, we'd soon have a very consistent PCI API again."

10 | LXF311 February 2024 www.linuxformat.com

Answers

Got a burning question about open source or the kernel? Whatever your level, email it to **answers@linuxformat.com**



Neil Bothwick is thinking about changing his name to Tux.

Access all areas

I've recently retired and now I'm learning about Linux and have installed Linux Mint on a spare laptop. I used to write MS Access database projects for the office where I used to work. I'd like to create some projects for myself and friends, and there was a feature that Lused back then. Access allows you to store the data and tables on a server and distribute a front-end app of forms, queries and reports that users could use on their PC to access the database, basically a client-server app. I was wondering if you know of any Linux database apps that work in a similar manner. I didn't see anything like that in LibreOffice Base and that's the extent of my knowledge of database applications in Linux. Can you point me to a software source that has that feature or is there an extension in LibreOffice Base that can be added on that mimics that feature?

Lupe Flores

Most database systems used on Linux are client-server arrangements and LibreOffice can be used as a client in this way. However, LibreOffice is only able to connect to an existing external database, it cannot create one. The most popular database engine on Linux is MySQL. This is the

server but it also includes command-line programs for creating, querying and maintaining databases. However, you may want a graphical tool to create databases and a couple spring to mind. MySQL Workbench is part of the MySQL project and should be in your distro's software repositories. Alternatively, the web-based phpMyAdmin (www.phpmyadmin.net) lets you work with MySQL databases in a web browser. You'll need a web server installed, too (usually Apache), but your distro's package manager should take care of that if you install phpMyAdmin through it.

Whichever way you create the database, you can then access it through LibreOffice. When you create a new database project in LibreOffice, pick the option to Connect To An Existing Database. Select the database type from the drop-down, say MySQL, and press Next. Then you are asked to choose a connector - this is the software that provides the connection between LibreOffice and the database server. The three options are ODBC, which has a reputation for being problematic, JDBC, which uses Java and is the default, or Connect Directly. The last option uses the built-in connector code in LibreOffice and is a good choice.

Then you can enter the database details: the name of the database and

the server hosting it (the port number should not need changing). Finally, give the username and password, and press Test to check the connection. Then you can open and save the database and work on it in *LibreOffice* from here on.

Ongoing destruction

I've just read your Answers section in the December 2023 issue of Linux Format. All looks good, thanks. It would be even better, however, if some more aspects of data destruction got covered. Such as how to overwrite the free space on a partition or how to build in Nemo (Nautilus) support of the Shred program in the right-click menu.

Vic

You can overwrite the free space on a partition with dd, using either zeros or pseudo-random data:

\$ dd if=/dev/zero bs=4M of=somefile or:

\$ dd if=/dev/urandom bs=4M of=somefile

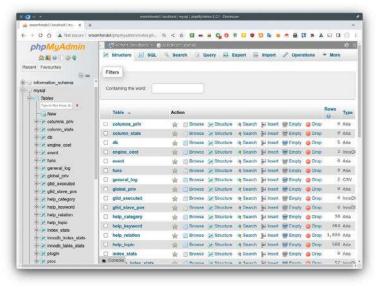
This creates a file in the current directory and continues until there is no more space, at which point it exits with an error. Now you can either delete or shred the file you just created, depending on the level of security you want when retrieving the free space. You can also use this approach when making whole partition backups or images using compression. Using the /dev/zero option and then deleting the file means that all unused space contains zeros, making the backup more compressible.

To add an action to the context menu in *Nemo*, you create a suitable file in ~/.local/share/nemo/actions with a .nemo_action filename extension, say shred.nemo_action, containing:

[Nemo Action]
Name=Shred file
Comment=Shred file
Exec=shred "%F"
Selection=any
Terminal=true

Once done, you should see the modified context menu. If you do not, log out of the desktop and log back in, or kill

You can manage MySQL databases from the comfort of your web browser with phpMyAdmin.



www.techradar.com/pro/linux February 2024 LXF311 |11



ANSWERS

all *Nemo* processes, to make sure the new configuration is used.

We also received an email asking if it was necessary to use a specialist tool like Shred - could you not simply overwrite the file with another file using the Save As function of a program? Shred is a specialist tool; Linux contains many such tools designed specifically for one job. It is part of the coreutils package, so is included by default on just about any Linux distribution you can think of. The problem with trying to overwrite a file with another is that although the new data may be saved with the same name as the old file, it may not be saved in the same place. So, even though accessing the old data by filename is not possible, it is still there and able to be read by forensic and repair tools, such as Photorec.

Unicode provides
a ridiculously
large number
of characters,
and you can
type them all
in LibreOffice
- you just need
the codes.

				Offi	cial Un				nbols		PDF)					
	0	1	2	3	4	5	6	7	8	9	A	В	С	D	E	F
U+260x	*	•	-	8	di.	*	命	<	ĸ	0	ស	ಆ	ď	00	8	100
U+261x		Ø	×	X	XBs.	0	Ω	•	*	.85	•	•	70	4	Dr.	19
U+262x	R	2	•	*		+	†	¥.	t	+	C.	40	ф	æ,	0	•
U+263x	Ξ	#	Ξ	Ξ	=	=	噩	Ħ	泰	8	0	۵	*)	•	ğ
U+264x	9	đ	ð	의	h	ж	Ψ	5	0	0	0	0	0	@	0	•
U+265x	0	0	0	0	ф	8	I	Á	63	A	*	讏	X	ù	4	A
U+266x	ė	Ø	0		۵	٧	٠	4	25	1	2	J	л	b		111
U+267x	+	+	43	۵	25	B	a	a	a	Δ	۵	٥	•	٥	@	3
U+268x	⊡		Ø	=	×	П	0	0	•	•	-		=	=	==	::
U+269x	þ	Þ	*	ů	×	1	55	A	4	o	Ť	8 0 0	*	\$3	7	=
U+26Ax	Δ	4	ф	ම	Φ,	ď,	उँ	Ş	ð	00	0	•	0	0	010	0-0
U+26Bx	0	٥	P	2	Ŷ	*	*	ŧ	\$	*	¥	π	Q	(8)	20	0
U+26Cx	0	8	0		塘	٥	D	0	8	0		0	0	0	0	15
U+26Dx	0	9		X 0	0	0	0	0		.0		0	D	0	0	0
U+26Ex			ô		0	0	D			Ħ	AOA					0
U+26Fx	٨	10)	32		A.	à	П	*	2	A.		П	П	-	П	0

More digits please

I am using Linux Mint 20 and LibreOffice. I recently discovered how to use Ctrl+Shift+U+[four-character hex code] to insert fancy icons into my documents. So delighted with this, I made a list of useful Unicode characters in my Linux crib notebook. But I then discovered that there are more Unicode characters beyond hex FFFF. But when I use five-character hex codes – for example, "globe" 1F310 – no character appears. Is there a file somewhere in my system that needs to be updated to allow me to have even more fun with Unicode characters?

Pete Barrett

Originally, LibreOffice only supported adding Unicode characters by code in its Linux versions, Windows users had to install an extension to do this. Later on, the facility was added for Windows users, but working differently. Interestingly, we tried with two

versions of *LibreOffice*, 7.5 and 7.6, on two different distros, and Ctrl+Shift+U followed by the code and Enter, did not work with either. However, the method implemented for Windows did work, including with five-digit codes. So this appears to be the correct way of doing things now and you will have to retrain your muscle memory. The good news is that the new method requires fewer key presses – simply type the code and then press Alt+X. Note that this is the other way round – you type the code first and then the key combination.

This works with all the codes we tried but, and this is stating the obvious for the sake of completeness, it only has an effect if the font you are using contains the extended Unicode characters.

No GUI, no cry
How can I boot a freshly updated
AMD64 Mint without a GUI? I want to
run a server and save resources. I already

tried to change the GRUB configuration to text mode without success. In the days before *Systemd*, I would edit /etc/inittab, but I am lost with *Systemd* because the old levels of 3 for text, 5 for GUI and so on are gone.

Eve Stephens

The runlevels are still there in Systemd, but they're called targets and have somewhat more descriptive names. With no other instructions, Systemd boots to the default.target, but this is just a link to your chosen target. On a desktop this is graphical.target while for a console boot you want multi-user.target. There is also single.target but this just boots the system and starts virtually nothing. Multi-user.target contains networking, system daemons, everything except the desktop really.

There are three ways you can switch targets. To do it interactively, run:

\$ sudo systemctl isolate multi-user.target

» A QUICK REFERENCE TO... ZSTD

A good compression rate, quickly achieved, is the holy grail of compression software. You may have noticed more files available with the ZST extension. These are produced by the Zstd compression program, one that is currently in vogue. In use, it sensibly works very much like the alternatives such as Gzip, Bzip2 and Xz. If you pass it a file, it compresses it with the

default settings and creates a new file using the original name plus .zst. Run it as unzstd (or zst-d) and give it a previously compressed file and it uncompresses it. There is one significant difference in the way it completes these operations in comparison with its precursors: it does not delete the original file. While Bzip2 and friends delete the file unless you use the -k/--keep switch, Zstd

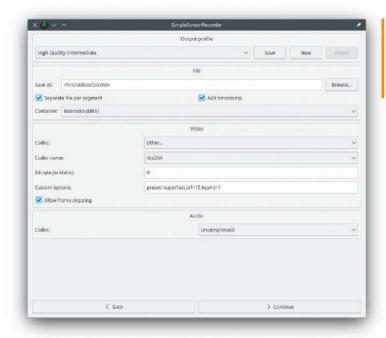
keeps them unless given the --rm switch.

The default settings give fast compression and decompression, but this can by changed by adding -N to the command, where N is between 1 and 19. The higher the number, the more aggressive the compression and the longer it takes. The default is 3, but there are diminishing returns when aiming for the higher reaches

of compression. There is also an --ultra option to unlock even higher compression levels, at the expense of much greater memory usage – both for compression and decompression.

The man page details all the available options and you could spend hours benchmarking the results, or you could stick with the defaults and just use zst and unzst – it is up to you.

12 | XF311 February 2024 www.linuxformat.com



Simple Screen Recorder can record your videos where (and how) you want, but it is not always the best tool for a particular job.

This starts everything required by the given target and stops all services that are not used by it, the desktop services in this case. You can make this the default when booting with:

\$ sudo systemctl set-default multi-user. target

This changes the destination of the symlink at /etc/systemd/system/default. target – much of Systemd's configuration is achieved by manipulating symlinks. The third way is to explicitly specify the target as a kernel option in the GRUB menu. Add this to the kernel options in the menu:

systemd.unit=multi-user.target

The advantage of this approach is that you can have separate menu entries that specify **multi-user.target** or **graphical. target** – or give no target and rely on the default setting.

On the record

I am a new user to Mint. Since I do a lot of video downloads for music off YouTube, I downloaded the Simple Screen Recorder. But it only saves a video file in one place: in Videos on my hard drive. My SSD is only 240GB so that will fill up soon. Therefore, I would like to download them to my SSD external hard drive, which is two terabytes. Do you know of a program on Flatpak that would do that? There is no good way to move the files over to the external drive and whenever I try to copy and paste, the file gets corrupted. Drag and drop is not an option either.

David Spencer

A There are a few issues here, but the first one to tackle is why are files corrupted when copied to your external drive. Are all files corrupted or just large video files? Using a screen recorder can

produce very large files and some filesystems, particularly FAT, have a size limit. Trying to copy a file larger than this appears to work, but the result is corrupted. So check that your hard drive is formatted with a Linux-friendly filesystem. If you need cross-platform compatibility, format it with NTFS and install the ntfs-3g package from the software manager to give the best results.

Changing the output location for Simple Screen Recorder can be done in one of two ways. Firstly, you can select a file in a different directory from the GUI by pressing Browse next to the filename box. You can also change the default output file in the configuration file. Load .ssr/settings.conf into your preferred text editor. In the [output] section, you will find a file option; set this to where you want to save the files, such as:

file=/media/exthd/ssr.mkv

However, Simple Screen Recorder is not the best tool for this task as it records the whole desktop or browser window in a format that produces large files. You are probably better off with a dedicated tool such as Yt-dlp (https://github.com/yt-dlp/yt-dlp). This is a fork of the original Youtube-dl tool which, as its name implies, is designed for downloading YouTube videos – although it works with other sites, too. It is a command-line tool but very easy to use. Copy the link to the video you want to download, then pass it to Yt-dlp in a terminal but pasting it, like this:

\$ yt-dlp https://www.youtube.com/ watch?v=xxxxxxxxx

If you are downloading from YouTube, you can do away with the full URL and just use the video's identifier, the part following v=:

\$ yt-dlp xxxxxxxxx

Spot the difference
A PC has two network cards, both
of which are the same model. How do
I find out which MAC address belongs
to which network card when it is not
possible to open the PC case?

George Dyer

An interesting problem as using Ispci-k will not help if both cards use the same module. Are these PCI network cards or ports built into the motherboard? If the former, the predictable naming scheme now used for network cards may help, as interfaces are numbered sequentially, according to their PCI slots. You really should be using this form of naming with multiple Ethernet ports, to ensure that the ports get the same name each time – the old eth0/eth1 naming does not guarantee that.

Then your question boils down to: you have two ports, say eno1 and eno2 for onboard interfaces or enp1s0 and enp2s0 for PCI cards, so how do you tell which is which? You can do that with *Ethtool*:

\$ ethtool --identify 60 eno1

This blinks one of the LEDs, usually the link LED, of the given interface for 60 seconds. Repeat it for each network interface and you will know which is which. Now get the MAC address of a specified interface with ifconfig or ip:

\$ ip link show eno1

Or list them all with just:

\$ ip link show

Once you have done that, the tried and tested method of writing the information on sticky labels comes into play.

GET HELP NOW!

We'd love to try to answer any questions you send to answers@linuxformat.com, no matter what the level. We've all been stuck before, so don't be shy. However, we're only human (although many suspect Neil is a large language model), so it's important that you include as much information as you can. If something works on one distro but not another, tell us. If you get an error message, please tell us the exact message and precisely what you did to invoke it.

If you have, or suspect, a hardware problem, let us know about the hardware. Consider installing *hardinfo* or *Ishw*. These programs list the hardware on your machine, so send us their output. If you're unwilling, or unable, to install these, run the following commands in a root terminal and send us the **system.txt** file, too:

uname -a > system.txt lspci >> system.txt lspci -vv >> system.txt

February 2024 **LXF311** 13

CREDIT: Alexey Komarov, CC BY-SA 4.0 Deed, https://en.wikipedia.org/wiki/PDP-1

Mailserver

WRITE TO US

Do you have a burning Linux-related issue that you want to discuss? Write to us at *Linux Format*, Future Publishing, Quay House, The Ambury, Bath, BA1 1UA or email **letters**@ **linuxformat.com**.

I think we've misplaced the PDP-1 in the Linux Format server dungeon.

Catch the MUMPS

Linux Format has recently been looking at coding languages from older systems. MUMPS was developed for the PDP-1 in about 1966-1967 at Massachusetts General Hospital.

MUMPS (which stands for Massachusetts General Hospital Utility Multi-Programming System), or M, is an imperative, high-level programming language with an integrated transaction processing key-value database. It was originally developed at Massachusetts General Hospital for managing hospital laboratory information systems (details taken from Wikipedia).

The main difference between MUMPS and other programming languages is the embedded key-value database. The last two or three implementations are *Intersystems Cache, FIS-GTM* and *YottaDB*. It is an ISO standard (ISO/IEC 11756:1999, re-affirmed on 25th June 2010).

Here's hoping you will do an article on this language in the near future.

Wally

Neil says...

Someone wake Mike up – we've got another language for him to fire up!

What flavour?

I have an old Advent 4211 laptop, which is a rebrand of the MSI Wind U100. It has 2GB of 667MHz RAM (the maximum it can take), an Intel Atom N270, a GMA 950 at 600p resolution, and a 945GC chipset. It is quite weak, but otherwise is fully functional. I am currently running Windows 7 on it, which while not being a terrible experience, is not the best, being old, having a lack of support – with less available every day – and not being the lightest operating system for this little laptop.

I tried Batocera on it, which was a great experience, and it now doubles as a retro gaming laptop. I also tried AntiX Linux on it, which works, but I think there might be other options. I have been recommended Mint, Lubuntu and Haiku, which I will all test, but I feel like these are not the lightest options. I am also thinking of getting Android or ChromeOS on it.

So, basically, what I want is something that is lightweight, has support for modern applications and security updates, and is relatively simple and user-friendly, because I am not the most experienced Linux user.

Brian Eastman

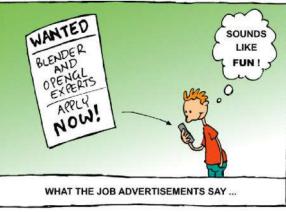
Neil says...

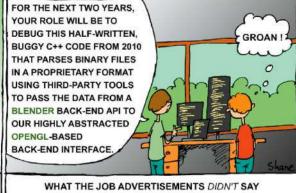
That's a 2008 Atom processor and is the real sticking point for you, because it's only 32-bit, which kills many modern options. It only supports SSE3, too, but that's less of an issue for general software, though it can be for modern games. Be aware that this is an old single-core Atom (although it does two threads via HyperThreading), and it only handles in-order command execution, so really is on the slow side.

AntiX is an excellent suggestion, although we also recommend you



Helpdex





14 LXF311 February 2024 www.linuxformat.com



We need to take a fresh look at the newly released Bodhi 7.0.

try Bodhi Linux (www.bodhilinux.com), which has a specific 32-bit Legacy build with frozen kernel for older hardware.

I suspect Mint and Lubuntu could be a bit heavyweight. Haiku could work, but it's just less widely supported than Linux. While Android x86 should be OK, I've always found app use to be odd. Google is offering Chrome Flex and there are ChromiumOS builds, but I believe both are 64-bit only.

Scanners

Mike Bedford hit the nail on the head for me in his article Capture And Restore Old Photos And Slides (LXF304) because I have long been looking for a way to digitise and restore my many slides and photos without having to spend a fortune to do so. Would Mike be able to recommend a few easy-to-use and relatively inexpensive scanners for our consideration? I would hope that Mike could continue along this line and make a series out of this article by going into more detailed explanations of the process for slides, negatives and photos. (I want more of his knowledge, please!)

Anyway, keep up the great work, because each issue keeps me out of my wife's hair for a good while!

Frank Gunseor

Neil says...

I'm glad you're enjoying Mike's articles. We'll try to keep the more accessible tutorials on the books for everyone to enjoy. As for recommending scanners,

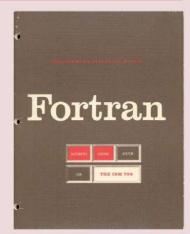
they're not the sort of thing we review

>> LETTER OF THE MONTH

Moretran

I think it may have been worth mentioning how the name Fortran arose, although I agree most of your readers probably already know what it stands for – FORmula TRANslation. It was the go-to language for turning mathematical formulas and algorithms into code.

It should also be added that, initially, little regard was paid to ergonomics for average users with its rigid input formats. Its main goal was to provide an efficient means of getting the most speed from machines.



Any excuse to print old manual covers...

Proprietary extensions to the language became commonplace between its 66 and 77 versions, which were standardised by the so-called X3J3 Committee on Languages, with the aim of easing some of these issues and additional changes, so that more checks could be made at compile time (for example, to avoid spelling errors, IMPLICIT NONE).

I used Fortran II, IV, 77 and 90/95. It seems to have 'rebranded' itself to fight back against other languages and remain relevant to today's computing, at the risk of making the language somewhat harder to learn than it once was. (I used it for most of my career and retired nearly 20 years ago.) In a similar way, Algol 68 was introduced to do a similar job for Algol 60.

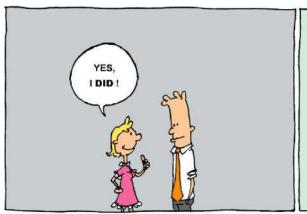
John Greenaway

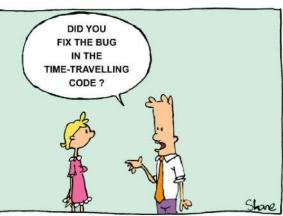
Neil says...

Thanks for the insight – it's all really very interesting. Keep letting us know!

I Thankfully not the Cronenberg type...

as such, but most have open protocols, as we looked at back in LXF248. The SANE (www.sane-project.org) project lists support for over 2,000 of them. However, I can also direct you to our testing experts at TechRadar: www.techradar.com/news/best-scanners.





shane_collinge@yahoo.com

SUBSCRIBE TO LINUX FORMAT MAGAZINE

Subscribe today and enjoy great savings!

www.magazinesdirect.com/LIN/C75J
Call 0330 333 1113 and quote C75J





Developer issues

PLUS: Exclusive access¹ to the Linux Format subs area!

1,000s of DRM-free PDF back issues and articles! Get instant access back to issue 66 (May 2005) with tutorials, interviews, features and reviews. At linuxformat.com

deals!

>> CHOOSE YOUR PACKAGE!

Our simple six-month best offer breaks up the annual subscription into two easily manageable payments over the course of the year.





Terms and conditions: Offer closes 29/02/2024. Offer open to new UK subscribers only. Pricing is guaranteed for the first 12 months and we will notify you in advance of any price changes. Please allow up to six weeks for delivery of your first subscription issue (up to eight weeks overseas). The full subscription rate is for 12 months (13 issues) and includes postage and packaging. If the magazine ordered changes frequency per annum, we will honour the number of issues paid for, not the term of the subscription. For full terms and conditions, visit www.magazinesdirect.com/terms. For enquiries please telephone +44(0)330 333 1113 (open Monday-Friday 8:30am–7pm, and Saturday 10am-3pm UK time), or email help@magazinesdirect.com. Calls to 0330 numbers will be charged at no more than a national landline call, and may be included in your phone provider's call bundle.

February 2024 LXF311 17 www.techradar.com/pro/linux

EXPLORE THE BEST LINUX & OPEN SOURCE SOFTWARE

From the best distros to system security, utilities and fun applications, we've rounded up everything you could ever need



FUTURE **m**

Ordering is easy. Go online at:

magazinesdirect.com

Or get it from selected supermarkets & newsagents

REVIEWS

Intel Core i5 14600K

No one calls Jacob Ridley mid-range!

SPECS

Socket: V LGA1700 **Process:** Intel 7 10nm Cores (P+E): 6+8 Threads: 20 P-cache: 480KB L1. 12MB L2, 24MB L3 (shared) P-core: 3.5GHz (5.3GHz boost) E-cache: 786MB L1, 8MB L2, 24MB L3 (shared) E-core: 2.6GHz (4GHz boost) **Unlocked:** Yes **GPU:** Intel **UHD 770 GPU** clock: 300MHz (1.55GHz max) Ex units: 32 Display: 4, eDP 1.4b, DP 1.4a, **HDMI 2.1** API: OpenGL 4.5, OpenCL 3.0 Mem max: 192GB, ECC support, 2-channel Mem speed: DDR5 5,600MT/s, DDR4 3,200MT/s **PCIe:** v5 & v4 20-lanes **Base power:** 125W

Maximum

181W

turbo power:

he K-series Core i5 processors are usually the easiest recommendation from any new Intel generation, but for once we've found ourselves less sure. On the one hand, this new chip is a couple of frames faster than the Core i5 13600K that it replaces and it does remain extremely competitive versus AMD's mid-range processors. On the other hand, it's a touch more expensive than the older 13600K and paying extra for near-enough the same thing doesn't feel particularly great to us. However, that's mostly because the 13600K is now discounted on its way out the door.

Due to Intel's decision to stick with a largely unchanged Raptor Lake architecture for the 14th generation - the same architecture that was used throughout most of the 13th gen - there are few differences between the Core 14600K and 13600K. And that includes the launch price.

The Core i5 14600K features six Performancecores (P-cores) and eight Efficient-cores (E-cores). That's a total of 14 cores with 20 available threads. With 24MB of L3 (Smart) cache and 20MB of L2 cache - 2MB for every P-core and 4MB for each of the two E-core clusters - it's stacking up as an exact match for the 13600K.

Picking up speed

The 14600K is slightly faster than its predecessor, however. Not in terms of overall highest clock speed, though - this newer chip is rated to 5.3GHz max boost, exactly the same as the 13600K - but the P-cores are up to 200MHz faster and the E-cores are 100MHz faster.

Most often, these faster clocks amount to a small increase in frame rates over the 13600K. That's a frame or two on average. There are still games, however, where the 14600K doesn't benefit at all from its minutely faster clocks. Even a frame or two is within reasonable testing variance ranges.

In synthetic benchmarks, there's even less to report from our testing. The Core i5 14600K holds a perfunctory lead over the 13600K, which amounts to no serious real-world benefit that you would notice in use. You can also expect a slightly higher power consumption on the 14600K. We put it at around 10W higher peak wattage during testing.

But this isn't a one-horse race. The more interesting fight is between the 14600K and AMD's lineup. You've got the Ryzen 7 7800X3D at £350, which comes with 3D V-Cache to boost frame rates in games that benefit most from more cache close at hand, or the cheaper Ryzen 7 7700 at £335, which doesn't.

The Intel 14600K and AMD 7800X3D are in a dead heat across the games we've used, with each



Look familiar? Well, perhaps that's because it largely is...

processor having three titles under its belt. When AMD's 3D V-Cache does work, however, it really does work wonders.

As for the 7700, this chip doesn't really come close to the 14600K for the same money.

In synthetic performance benchmarks, it's Intel's higher overall core counts and strong P-core performance that help it maintain a solid lead. In Cinebench R23, Intel's mid-range chip absolutely dominates AMD, and even in encoding or rendering benchmarks, such as x264 or Blender, the 14th-gen chip pulls ahead.

The downside of the 14600K is that it's a whole lot more power hungry than AMD's Ryzen 7 - consuming double the wattage at peak load.

Intel does net the benefit of being a cheaper upgrade for a builder with an LGA 1700 motherboard already in their machine. The 14th-gen is compatible with any 600 or 700-series motherboard, marking the third chip generation to use the same socket. So, against AMD, the 14600K is the one to choose; against the 13600K, however, if you can find that chip cheaper, that's the one to buy.

VERDICT

DEVELOPER: Intel **WEB:** www.intel.com PRICE: £320

FEATURES	9/10	EASE OF USE	9/10
PERFORMANCE	8/10	VALUE	8/10

Not as easy a recommendation as the Core i5 13600K. Yet this is a great value pick versus AMD's 7000-series chips.

Rating 8/10

February 2024 **LXF311** 19 www.techradar.com/pro/linux

EndeavourOS 11-2023

Nate Drake endeavours to discover all that is great about this Arch-based Linux distro, created by former Antergos developers.

IN BRIEF

EndeavourOS is definitely a more user-friendly version of Arch Linux. Installation is a breeze, with an array of desktop environments and window managers. Updates/extra software are handled by integrated tools.

SPECS

CPU: 1GHz (dual-core) Mem: 2.5GB HDD: 15GB Builds: x86_64, ARM64 ack in May 2019, the developers of Antergos, an Arch-based Linux distro, announced that the project would no longer be maintained. Disappointed, Antergos forum users banded together to continue the project under a new name, thus EndeavourOS was born that July.

If you're already a user, there's little to do given that EndeavourOS follows a rolling release model. Just run the integrated *Eos-update* tool to enjoy the latest Galileo version.

If, however, you download the 2.7GB ISO, you can also test the OS in a live environment. Unlike Arch Linux, which uses a command-line tool for setup, EndeavourOS

deploys the *Calamares* installer, which you can launch from the helpful welcome window, which also contains options to load the EndeavourOS ARM image installer, update software mirrors and manage partitions.

If you proceed with installation, you're offered two setup methods. Click Online to choose from a variety of desktop environments, including LXQt, Budgie, Cinnamon, Xfce and Gnome. The offline installer loads the desktop environment that comes with the disk. In the latest version of EndeavourOS, this is now KDE Plasma with some distro-specific branding.

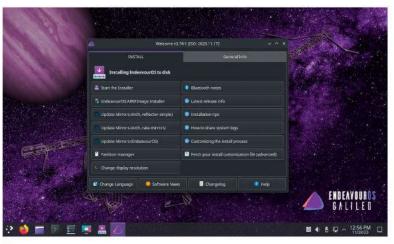
Due to a lack of support from developers, the community editions of *Sway, Qtile, BSPWM, Openbox* and *Worm* are no longer available via *Calamares* but can be installed manually via GitHub.

Whichever install method you use, the *Calamares* package selection screen has been restructured to be clearer and to make some items more discoverable. When LUKS encryption with *Systemd-boot* is chosen, the system is also installed with a stronger LUKS2 encryption using Argon2id.

The latest version of EndeavourOS has been slimmed down to make it more efficient. A bug that previously generated a superfluous LUKS keyfile has also now been fixed. Former versions of the distro also generated numerous SELinux warnings during install. These have now been removed to avoid confusion.

The installer now only allows users to install one desktop environment and a single window manager. The developers' stated aim is to avoid package conflict, although additional desktop environments and window managers can be added manually post-install.

At the time of writing, one major bug persists in the installer: if you select empty, unpartitioned space or an empty partition with an unsupported filesystem, *Calamares* won't display a warning before overwriting



Endeavour0S's new welcome screen offers a partition manager, mirror configuration and language options.

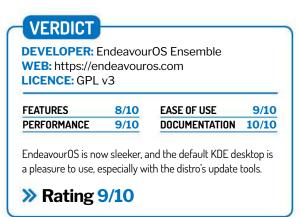
it. The developers point out you can overcome this by using the partition manager in the *Welcome* app.

Under the hood, EndeavourOS uses version 6.6.1 LTS of the kernel. The aforementioned *Eos-update* tool now has a new --keyrings-reset option, to help with corrupted keyrings. There are also new mirrors around the world, to make updates and installs speedier.

One of the most impressive features of this OS is the level of online support available. Its comprehensive wiki contains a dedicated installation guide (https://discovery.endeavouros.com/category/installation), which includes tutorials on using Systemd-boot, adding packages to be installed alongside the default desktop, as well as creating install media.

The main wiki also points to a number of YouTube channels with step-by-step guides on configuring your EndeavourOS install and general troubleshooting. The distro also has a dedicated forum, as well as its own Twitter, Mastodon and Telegram channels.

Overall, the stunning graphics, easy setup and high level of online support make Galileo one to watch.



20 LXF311 February 2024 www.linuxformat.com

SysLinux0S 12.2

Nate Drake dives into SysLinuxOS to find whether its reputation as the dream OS for system administrators is well deserved.

IN BRIEF

This distro has everything you're likely to need for managing networks and systems. Both Gnome and MATE versions are available. The tools are also plentiful and well categorised.

SPECS

CPU: 1GHz (dual-core) Mem: 2GB HDD: 20GB Builds: x86_64 ot to be confused with SysLinux, a bootloader for Linux that can run on Windows partitions, SysLinuxOS is a Debian-based GNU/Linux live distribution primarily designed for network administrators and system integrators.

Visitors to the main website are offered a choice between versions of SysLinuxOS running the MATE and Gnome desktop environments. The latest release (12.2) is based on Debian 12 Bookworm.

The website also boasts that there's a "Swiss army knife" of tools, such that integrators and administrators have no need to install anything extra. After

downloading the 4.5GB ISO, we took the Gnome version of SysLinuxOS for a test drive.

Upon first boot, we saw that the heavily-sized ISO was justified. Tools in the dock include administration stalwarts such as *Terminator*, *PuTTy*, *Team Viewer* and *Packet Tracer*.

The main website also mentioned that SysLinuxOS supports "all the major VPNs", so we were keen to put this to the test. But running a search via the Activities window, we only discovered software for RiseUP VPN.

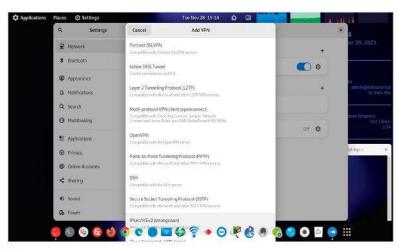
We fired up the *Firefox ESR* browser to check if any VPN extensions had been preinstalled, but had no joy. When checking network settings, however, we found the configuration manager supported a number of protocols, including IPSec/IKEv2 and Fortinet SSL VPN SSTP, so assume that the developer meant that SysLinuxOS supports all major VPN protocols out of the box. The one notable exception was Wireguard, which while listed on the SysLinuxOS website, didn't appear in Network Settings.

The developer's claim that the OS supports "several remote control clients" was much easier to verify as apps such as *TeamViewer* and *AnyDesk* can be launched directly from the dock. *Remmina, Zoom* and *Skype* also come preinstalled.

If you do take the time to explore the installed applications, you'll see that SysLinuxOS helpfully groups the wide array of programs into categories such as Firewall, Internet and Networking.

The latest version of SysLinuxOS comes with a number of updates, including supporting booting from a USB. The OS now uses version 6.5 of the Linux kernel and if you do choose to install to a disk, this is handed by the intuitive *Calamares* installer.

Although the Firefox icon is nestled in the dock, SysLinuxOS 12.2 also now includes Google Chrome



SysLinuxOS supports a number of VPN protocols, but despite the main site's claims, Wireguard didn't seem to be listed.

and Microsoft Edge alongside it. You can also launch the Tor Browser and Chromium via Activities.

Admins will be very familiar with the finely curated selection of networking tools including *Wireshark*, *Packet Tracer 8.2.1*, *GNS3*, *Nmap*, *Lssid*, *Etherape*, *Ettercap* and *PackETH*. The OS also includes *Docker*.

We were a little surprised at the inclusion of Virtualbox 7.0 and Vmware Player 17 but imagine they have their uses for malware testing. SysLinuxOS also comes with Wine for running Windows apps, as well as FileZilla for easy remote access to servers.

If you speak Italian, you can benefit from Edmond's Weblog Tutorial, which details how to update and configure built-in tools like *Sparrow WiFi*. There's also a dedicated SysLinuxOS forum (once again in Italian), but it doesn't seem to have been updated in a while.

Still, as the developer notes, "This distro is mainly oriented for those who have excellent skills in the Linux environment." This implies SysLinuxOS is aimed at users who don't need much hand-holding when it comes to using its Swiss army knife of tools.



www.techradar.com/pro/linux February 2024 LXF311 | 21

OpenMandriva Lx 5.0

Nate Drake relives Mandriva Linux's heyday with its programmer and community-developed successor. Plasma has never looked so good.

IN BRIEF

Whether you opt for the fixed or rolling release, you are in for a treat with OpenMandriva. Setup walks you through features and you can configure every aspect of the Plasma desktop.

SPECS

CPU: 1GHz Mem: 2GB HDD: 10GB Builds: x86_64, znver1 any Linux users have fond memories of Mandriva Linux, an RHEL-based distro that was discontinued in 2011. OpenMandriva was born from its ashes through the efforts of former Mandriva developers and the coding community.

The first version of OpenMandriva hit the internet a little over 10 years ago, and since then, the OS has consistently received positive reviews for its broad appeal and ease of use.

Traditionally, the OpenMandriva project has concentrated on fixed releases, such as Iodine Rock 5.0, which is the focus of this review. However, in January 2023, the team

also released ROME, a rolling release that contains the latest stable version of all OpenMandriva packages.

The current version of OpenMandriva uses the KDE Plasma 5.27.9 desktop, the last distro to do so before the upgrade to KDE Plasma 6. The OS also includes KDE Gear 23.08.3 and KDE Frameworks 5.112.0.

ISO downloads are offered as an Extended version, which comes bundled with software for most day-to-day tasks and weighs around 3GB, and a Slim version containing the bare essentials, such as the *Dolphin* file manager and graphical package installer *Dnfdrake*.

Outside of component updates, this is the first fixed point release that merges the / and /usr filesystems, improving efficiency and compatibility.

At the time of writing, the distro is only available for x86_64 and znver1 builds, but there are plans for both ARM and RISC-V ports.

Current versions use version 6.6 of the Linux kernel. We discovered this on first boot through the OM Welcome screen. This is also where we discovered that HT/SMT NICE is enabled by default to improve performance on PCs that support hyperthreading.

The welcome guide also pointed us to OM Feeling Like, which enables you to configure the desktop to have a look and feel similar to other operating systems, such as Mac OS Mojave or Windows 10.

Checking the Global Theme in System Settings, we found that only the default OpenMandriva and Breeze themes are preinstalled, but you can download others.

As this version of OpenMandriva celebrates the OS's tenth anniversary, there's a huge parade of wallpapers, so you can choose from the very best OpenMandriva backgrounds from the past decade.

Special mention should also go to the new *OM Control Center.* The app contains various tools to help you configure aspects of your system such as software



To celebrate 10 years of OpenMandriva, version 5.0 comes with a selection of the very best wallpapers from the past decade.

management, disks and security. These are neatly categorised via sections along the top of the window.

In terms of apps, the welcome guide gives special mention to *LibreOffice*, now the most recent version (7.6.3.2), which comes with a number of bugfixes and small improvements, such as a page number wizard.

Other new programs included *Krita* for digital painting and the *Kamoso* webcam viewer. *Chromium* is still present. However, the OS also now comes with version 23.08.03 of KDE's own *Falkon Browser*, which includes excellent privacy features like the built-in AdBlock. The default search engine is DuckDuckGo.

OpenMandriva also bundles *KMail* and *KBackup* for handling emails and backups. The release notes indicate that media playback is handled by *VLC Media Player* but it didn't seem to be present in our tests. Still, this gave us a chance to pull up *Dnfdragora* to search for and install the app, which took under 20 seconds.

The welcome screen also made mention of Systemtap, a system performance analysis tool for the "technically adept". Invoke it with stap via Konsole.



DEVELOPER: OpenMandriva Association **WEB:** www.openmandriva.org

LICENCE: Free Software (mainly GPL)

FEATURES 8/10 EASE OF USE 9/10 PERFORMANCE 9/10 DOCUMENTATION 8/10

Excellent welcome guide, carefully chosen default apps and highly configurable desktop, making it ideal for general use.

>> Rating 9/10

22 LXF311 February 2024 www.linuxformat.com

Ultramarine Linux 39

Nate Drake decides to take this Thai Fedora-based distro for a spin and finds it fast, simple and surprisingly customisable.

IN BRIEF

If you like Fedora, you'll love Ultramarine's fine selection of desktop environments, easy setup and extensive documentation. Download today or upgrade from stock Fedora via the developers' own specialised script.

SPECS

CPU: 1GHz Mem: 4GB (8GB recommended) HDD: 10GB (20GB recommended) Builds: x86_64, ARM64 Itramarine is remarkable in that it's the only Linux distro we've encountered to come out of Thailand. The aim, according to the project wiki, is to be "a spiritual successor to Korora Linux, with the goal of making an operating system that 'just works'."

Korora was a remix of Fedora Linux and discontinued in 2017. True to its word, though, the current version of Ultramarine (code name Bears) is based on Fedora 39.

The OS is available with a number of desktop environments. The Flagship edition, which is the focus of this review, uses Budgie, and in the developers' own words,

includes their "latest and greatest software". It weighs in at around 2GB. Other versions are available with Gnome and KDE, as well as a Pantheon edition, which uses the default desktop of Elementary OS. Crucially, there's also now an ARM edition of Ultramarine, which can be flashed on to a Raspberry Pi.

Whichever version you choose, Ultramarine has undergone a revised build process that no longer relies on Red Hat's *Live Image Creator*. The team has used its own in-house builder, *Katsu*. Due to being written in Rust, it is not only lightning fast but allows better customisation of images, like those for Chromebooks.

Currently, installation images for Chromebook are available but only for more recent models. The team plans to use *Katsu* to bring Ultramarine to other Chromebook models in the near future.

The OS also now ships with a new default desktop background, created by the ever brilliant Aikoyori. There are five other wallpapers if you find the bundled treetop theme isn't to your taste. If you choose the Flagship edition, as we did, you'll also see the default typeface is now Inter instead of the previous Cantarell.

Ultramarine 39 has also now been updated to use the *Nemo* file manager instead of *Nautilus*. While we don't agree with the developers' assessment that *Nemo* is necessarily more powerful, it definitely matches the look and feel of the Flagship edition.

The OS also comes with a new in-house tool called *Stellar*, which automatically detects and installs Nvidia drivers during installation. You shouldn't need to do anything (except connect to the internet) for it to work.

The OS can be tested in a live environment. Users of Red Hat-based distros won't encounter any surprises if they proceed to set up, though, given that Ultramarine uses Fedora's own *Anaconda* installer.

Naturally, this begs an important question: why bother with this OS if you can already install stock



The Flagship edition of Ultramarine uses the Budgie desktop. However, there are also versions available that use KDE, Gnome and Elementary OS.

Fedora? While the installer is the same, Ultramarine does offer more than a fancy desktop background.

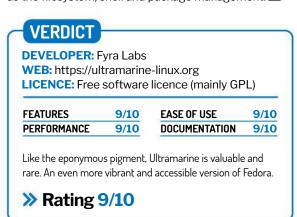
Chief among these is that IP laws work differently in Thailand, so this distro can ship with additional software above and beyond Fedora itself. Extra packages include version 7.6.2.1 of *LibreOffice*. The terminal also has an enhanced ZSH theme.

Perhaps most importantly, upon checking the repositories, we discovered that the RPM Fusion repois enabled for both free and non-free packages.

Fedora users who decide that Ultramarine is right for them can also upgrade via a dedicated install script.

Web browsing is handled by Firefox 120.0, designed for Fedora. This can be launched via the Flagship edition's Dock, along with Parole and Rhythmbox for media playback. You can install additional programs via Gnome Software. Packages are installed via Flathub out of the box.

Special mention should also go to the Ultramarine wiki, which has a dedicated Getting Started section, along with an explanation of key Linux concepts, such as the filesystem, shell and package management.



www.techradar.com/pro/linux February 2024 LXF311 | 23

Steam Deck OLED

Management is overjoyed: "Take an old product and slap a new badge on for more money?" **Tyler Colp** likes his job too much to correct them.

SPECS

OS: SteamOS 3.5 **APU:** AMD Sephiroth **Process:** TSMC 6nm CPU: Zen 2 Clock: 2.4GHz (3.5GHz boost) Cores: 4 Threads: 8 **GPU: RDNA 2, 8** compute units Mem: 16GB LPDDR5-6400 **HDD:** 512GB (option 1TB) NVMe SSD, microSD Screen: 7.4-inch OLED, 1280x 800.90Hz **HDR:** 1,000cd/ m² (600cd/m² SDR) Comms: Triband Wi-Fi 6E (ax) 2x2, Bluetooth 5.3 Battery: 50Whr **TDP: 4-15W Size:** 298x117x

49mm, 640g

f the Steam Deck was a beta, the Steam Deck OLED is your finished form. It's not a sequel, not a beefed-up 'pro' version, and not a major redesign. You can't even tell the difference between the old model and the new one when you sit them next to each other – save for the new orange accents. The Steam Deck OLED, which replaces most of the current models at the same price, is the ultimate form of one of the best handheld gaming PCs available.

The differences between the original Deck and the Steam Deck OLED are largely invisible, but they refine what was already a fantastic way to play PC games wherever you want. There are clear improvements with its most crucial stats: it has noticeably better battery life, more storage, a sharper, brighter screen, and weighs a tad less. But there are surprises, too: the touchscreen is way more responsive, the thumbsticks have an improved grip, and the bigger fan purrs much softer than before.

You won't want to go back to the original Steam Deck now. You don't hate it, but the OLED captures the promise of the original in ways you didn't initially expect. And while you don't think there's a sensible reason to upgrade if you've already bought one, the Steam Deck OLED cements Valve's device as the most accessible handheld gaming PC you can buy right now.

Your favourite games radiate on your new Deck's slightly larger 7.4-inch screen. The OLED panel can push 1,000 nits of brightness, so when you take your first steps into *Elden Ring's* pastoral opening region without properly adjusting your settings, it's like someone flipped on light mode. And with an impressive 110% of the DCI-



P3 gamut, the colours on your Deck OLED are deliciously vibrant. Given your Deck's modest resolution – it's still 1280x800 – the colour accuracy combined with the sharp 1,000,000:1 contrast ratio distracts you from the fuzziness of playing games below 1080p.

When it's on, HDR clarifies the extremes of a scene, revealing detail in the darkest blacks and brightest whites. Neon signs pierce through the dim bar in one of *Cyberpunk 2077*'s intros without clouding the shape of the letters with bloom, and the brilliant deserts in *Diablo 4* don't swallow your character whole. On such a small screen, that level of clarity meaningfully impacts fast-paced games. You can actually see what's going on as your sorceress electrifies entire rooms of demons. It's almost hard to go back to your regular PC monitor that is embarrassingly incapable of displaying that much detail.

Take charge

The cost, of course, is a little bit of battery life, depending on your brightness setting. The Deck OLED bumps the battery up to 50Wh (from 40Wh) for a noticeable improvement in the amount of time you have before you're tethered to a charging cable (which Valve has also improved, making it way longer).

Even if the impact is minor, you won't be trading away the Deck OLED's 90Hz mode. Spend years playing PC games at 144Hz and then pick up a 60Hz handheld and tell yourself it's not painful, at least until you finally give in and adjust. A screen that can't refresh itself fast enough to match the speed of a game adds blurriness to motion and a delay between your button inputs and the action. That's not true with the Deck OLED, which can take, say, a game locked at 30fps and triple each frame so it runs like butter at 90Hz. With a less than 1ms response time, games that waver around 30-40fps (that's many of them) have any hitches in responsiveness completely smoothed out. You can, of course, lower the target refresh rate in your Deck's overlay settings, which didn't significantly impact battery life in our test. You'll happily give up a few minutes to play on 90Hz all the time.

Not every Deck model has this power, though. As part of the Deck OLED refresh, Valve is only keeping

www.linuxformat.com



Peer inside the workings of the old and new Steam Deck – it turns out not to be steam-powered.

Game console **REVIEWS**



The raft of small improvements keep the Deck's position as the best handheld

>> BETTER BATTERY LIFE

At half brightness, HDR doesn't make a huge difference – at least when you have the choice. Software updates might change this down the line, but right now the Deck OLED defaults to HDR on for modern games. *Baldur's Gate 3*, for example, doesn't even let you turn it off in-game. That said, playing games at 90Hz with HDR on would only knock the battery life down by 20 minutes or so in our tests.

With default settings on the original Deck (60Hz, Wi-Fi on) and default settings on the Deck OLED (90Hz, Wi-Fi on), you can start to see what that new battery is doing. The Deck OLED pulled out nearly another 45 to 60 minutes of battery life compared to the original. Certain games, such as *Cyberpunk 2077* and *Baldur's Gate 3*, are going to eat up a lot of your juice no matter what, but squeezing out any extra time is a boon for a system like this. And if you play less demanding games, such as *Vampire Survivors*, you're looking at over an hour of extra battery life – or more.

the 256GB LCD model (the 64GB is available at £309) around and lowering its price to £349. A 512GB Deck OLED for £479 sits in the middle, next to a 1TB model for £569. The most expensive version also comes with a carrying case that hides another, thinner carrying case inside of it – we love it, but it's definitely not worth the upgrade alone.

There's a list of smaller improvements that don't make as much of a difference individually, but are worth mentioning to get a full picture of what it's like to use this thing daily. Here's what stuck out with regular use:

- You don't have to immediately reach for an SD card with how big the baseline SSD size is now.
- Touchscreen use isn't a laggy mess with its new 180Hz polling rate.
- The new thumbstick grips and smoother movement are great, like using an Xbox controller.
- Downloading games over Wi-Fi 6 rules and should have probably been there from the start.
- The weight reduction of about 29g is surprisingly noticeable when you can't rest your arms on anything.
- Not only does it keep a charge for longer, but it also charges faster now.

 You barely notice the fan noise unless you are actually listening out for it.

For comparison, we forced ourselves to swap back to the old Deck after playing on the new one for a week, and it was painful to look at its dull LCD display, rest our thumbs on the world's most slippery thumbsticks, and ride the roller coaster of gaming without the new Deck's refresh rate tech. But after 30 minutes, most of that friction wore away. You can't diminish the novelty of lying in bed or sitting on a sofa while you mosey around in *Baldur's Gate 3* or blast through a dungeon in *Diablo 4*.

For all the improvements, they're not so revolutionary that you can't stand using your old Deck. You miss the extra battery life and the screen the most, but unless you've got a buddy dangling the newer model in front of you, you're probably better off waiting for whatever Valve releases next. Because other than shifting some internal hardware around and shrinking the four-core AMD APU down a little, the Deck OLED runs the continuously growing list of Verified Steam games just the same. Valve has even said it will continue to maintain software parity with both models for the foreseeable future.

For those who haven't decided on a Deck yet, this is the absolute best version to get. Anyone with a big Steam library or the drive to tinker with Linux, there's simply no better handheld available. When we visited Valve to pick a Deck OLED up, almost every developer we met described the Deck OLED as the device they wanted to make from the beginning. It's the device you would have wanted from the beginning, too. At a price that still slips in under the competition despite all of its improvements, the Steam Deck OLED earns the crown as the king of handheld gaming PCs.

VERDICT DEVELOPER: Valve WEB: https://store.steampowered.com PRICE: £479 512GB (£569 1TB) **FEATURES** 9/10 EASE OF USE 9/10 **PERFORMANCE** 8/10 VALUE 8/10 While it may not push the performance higher, the OLED screen, improved battery and larger storage make it the most valuable device available. Rating 9/10

www.techradar.com/pro/linux February 2024 LXF311 | 25

WE COMPARE TONS OF STUFF SO YOU DON'T HAVE TO!

Roundup

Dolphin » Krusader » Midnight Commander » Nemo » Thunar



Michael Reed has written widely about Linux and open source software, and his files are always well managed.

File managers

File managers are the interface between the user and storage, and **Michael Reed** looks at five of the best that Linux has to offer.

HOW WE TESTED...

Our testing system this time was Linux Mint with the Cinnamon desktop. This detail is important because Linux applications such as file managers tend to inherit the theme and icon-set options of the underlying desktop environment.

Some of the features also rely on subsystems of the underlying distro. For example, network and archive mounting might rely on GVfs or KIO services on Gnome or KDE systems respectively.

For the archive testing, we created some example ZIP archives and downloaded some ISO images to test the systems. We made sure that we actually carried out some file management jobs to roadtest each file manager and throw up any awkward moments with each user interface or technical glitches once they were under the strain of actual work.

In the case of Midnight Commander and Krusader, we built from source to obtain the latest stable version as the versions in the Mint repository were out of date.



he file manager is what sits between you and the files on your storage devices. Many of us have a favourite, but it's just as common to make do with whatever is installed by default by your distro. Maybe we can inspire a switcheroo to something different. We're putting the 'big five' under the LXF microscope.

Dolphin is part of the KDE project and is an evolution from the earlier Konqueror web browser/file manager project. It's easy to use, even though it's a heavyweight in terms of the features that it offers.

Krusader comes from a similar KDE lineage, but it's a two-pane file manager

aimed squarely at tech-heads due to all the customisation and integrated tools it offers.

Thunar is the official file manager of the Xfce desktop environment, and it has a reputation for being light on resources yet offering most of the features a user is likely to need.

Midnight Commander is, like Krusader, a two-pane file manager with a lot of features, but it leaves out the GUI because it is a textmode utility.

Nemo is one of the forks of the original Gnome file manager Nautilus, and it aims to retain the feature set of older versions. So, let's see which you should consider installing. CREDIT: Getty Images/Brand X Pictures/Rubberball

Appearances are everything!

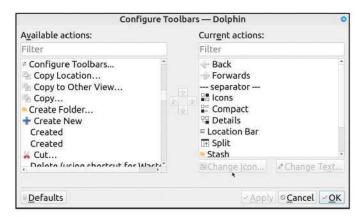
Tweaking the visible controls and changing how the interface works.

hunar has a good set of configuration options that control how it looks. Similarly, most areas of the user interface can be controlled in terms of icon and font size, and there are many smaller options for subtle tweaks. Thunar's layout options cover most of the basics you'd expect.

Nemo takes a similar road. The information below a file icon can be controlled, with three slots for this. The list view can have the number of information columns bumped up quite a bit. A few toolbar icons and context menu entries are disabled by default and can be brought into action. Nemo has the bare minimum of configuration options to remove or add something.

Dolphin starts with an understated interface that has a considerable capacity to be reconfigured. When you start to add toolbar icons, you discover that Dolphin has dozens of functions that can be assigned to a visible icon. Regarding file and directory options, there are quite a few choices about what is displayed. So, for example, you can specify that file and directory sizes are displayed as part of the icon view. All the file managers here can sort by different criteria but Dolphin offers particularly detailed options, such as sorting music files by artist or song.

Krusader is an absolute tweaker's delight, to the point where the sheer number of options can be overwhelming. Some of the configuration pages have multiple tabs. Making these options



Dolphin has a large number of extra icons that can be added to the toolbar. This is typical of the extra features that can be enabled in this file manager.

searchable would be welcome as there are so many. Certainly, if you can work within the two-pane model, there's little you can't add to the interface. Unlike the other GUI file managers, it's easy to alter the colour scheme rather than rely on the system one.

Midnight Commander sticks to a two-panel text-mode interface, but it does offer some options, such as a vertical or horizontal split for the panel, and exactly what is being displayed in the panels and what the format is.

VERDICT							
DOLPHIN	8/10	NEMO	6/10				
KRUSADER	9/10	THUNAR	6/10				
MIDNIGHT COMMANDER	6/10	\ <u></u>					

Dolphin and Krusader are the leaders in this section, although Krusader borders on giving too many options with a lack of organisation.

Behaviour configuration

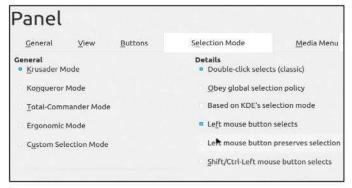
Removing a workflow glitch, or adapting the interface to suit you.

hunar offers a good, basic set of options for interface behaviour, such as switching to single-click launching of items. It's possible to specify that previously open tabs are restored on launch and you can add custom actions to context menus. *Thunar's* offering is the baseline of minimum features.

Pretty much every area of *Dolphin's* behaviour is configurable, extending to things such as making the Tab key switch between panels in multi-panel mode and specifying the effect of single and double clicks. Thankfully, date and time formats can be made absolute rather than "a few minutes ago" etc, which is the default. The context menu has extra options that can be activated and can connect to an integrated store to download new functions.

Krusader takes everything Dolphin offers and adds to it. Most aspects of what the mouse and keyboard do in combination have options. This is particularly important for Krusader as some users prefer to rely on the keyboard for quick and accurate operations.

Nemo covers the basic tweaks that most users will be looking for, such as restoring the previously open tabs on startup and



Krusader has a great number of small options that cover most areas of the user interface. It also has some presets from other file managers.

double or single-clicking to launch files. It's close, but *Thunar* and *Nemo* are probably neck and neck in this department.

Midnight Commander feels constricted to a certain way of working, but it does offer a few choices in terms of what key does what and how things such as multiple selection work.

VERDICT							
DOLPHIN	8/10	NEMO	6/10				
KRUSADER	9/10	THUNAR	6/10				
MIDNIGHT COMMANDER	5/10	8					

Krusader and Dolphin offer a lot of fine tweaking detail, but all of the GUI file managers offer the basics.

www.techradar.com/pro/linux February 2024 LXF311 | 27

The user interface

The interface between you and your files.

pplications like file managers tend to adopt the overall system theme and icon set of your desktop environment. When we fired up the fresh installations of Thunar, Dolphin and Nemo, what struck us was how similar they looked. Things only started to change when we began to configure and expand the user interface options.

Apart from colour scheme and icon set, layout is an important consideration. Thunar, Dolphin and Nemo use a similar layout by default, and it's probably fair to say that a number of standard conventions have established themselves in terms of how file managers tend to be operated and how they present the content to the user.

Simple or complex? We're not going to say that one is better than the other, as long as an interface achieves its goals. We want interfaces that neither are difficult to use nor make it difficult to understand what they are trying to display.

Dolphin

7/10

Krusader

8/10

We're assessing Dolphin in its default state, which happens to be a configuration with a lot of the features switched off. When first installed, the menu bar is accessible through a pulldown menu icon. This means that it's kept out of the way if you rarely access it, but awkward if you do want to use it. We don't see what the advantages are in keeping the toolbar as plain as Dolphin does on default. For example, it doesn't even have icons for up or home folder navigation, or a refresh button. It does have a split view icon, though.

The Places panel to the left works in much the same way as most file managers. Along the bottom, there is a status bar showing free disk space and the current zoom level. On first installation, Dolphin looks like most up to you to customise it in the areas that you care about.

other lightweight file managers, and it's 28 9° 02 Split Q

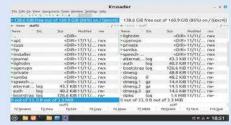
■ -

Krusader is a stablemate of fellow KDE project Dolphin, and this file manager is unashamedly aimed at technical users who like all of the features turned on combined with plenty of potential for both expansion and customisation. What you get with Krusader, before you embark upon customisation, is a dualpanel layout.

We've seen some dual-panel layouts that duplicate the available controls, but in Krusader's case, we're given a single toolbar that affects the currently focused panel.

Within a few minutes, you should be able to get the gist of how it works, because it's not a difficult user interface to pick up, even though we wouldn't give it to a non-technical user. It offers many of the advantages of a GUI file manager combined with the flavour of a textmode file manager. It's not a beginner's tool, and nor does it try to be.





Archive handling

It's common to have to work with archives in formats such as ZIP or ARJ.

hunar can support archive handling by means of a plugin that has to be installed manually. The plugin adds context menu entries to create and extract archives, and it works perfectly well.

Dolphin offers similar facilities on its context menus from a fresh installation. It can also open archives natively, but this needs to be enabled in a configuration menu. In our case, that didn't actually work until we did a reboot of the system. This is the archive handling method that some people will prefer, and it's particularly handy if you drop into two-pane mode.

Dolphin and Nemo have in common that they require that the underlying system support be in place for the seamless archive handling to work. On two different machines, we found that Dolphin couldn't successfully mount a ZIP archive, but it was able to mount ISO images. Forum posts indicated that this is an ongoing issue. From our experience, don't rely on seamless archive access within Nemo; on the other hand, context menu operations did work.

Krusader has context menu options for extraction and compression, and the archives open up as though they are ordinary directories when clicked on. Because it's always in two-pane mode, this is highly convenient. It can also launch an external tool such as File Roller to handle archives.

Much the same can be said of Midnight Commander as it also enters into archives as though they were directories. Pressing F2 invokes a pop-up menu with archive creation options. It worked fine on the test ZIP file, but it threw an error when accessing an ISO image, even though it did enter it correctly.

VERDICT DOLPHIN NEMO 6/10 9/10 **KRUSADER** 9/10 **THUNAR** 6/10

8/10 Archive handling within Krusader is consistent with normal file handling, while Thunar handles the basics.

www.linuxformat.com 28 **LXF311** February 2024

MIDNIGHT COMMANDER

Midnight Commander 7/10

If you like text-mode interfaces, you'll like *Midnight Commander*. The most common file functions are linked to F keys and displayed along the bottom. F9 takes you to the main menu at the top with more complicated functions. This keyboard-driven interface brings with it a few advantages in terms of precision and comes into its own for repetitive, accurate file operations and directory navigation. Similarly, when looking down a list of files, there is less room for ambiguity.

However, despite all of this praise, it is worth noting that the majority of people will struggle a bit to come to terms with something as alien as a text-mode interface if they were brought up in the world of the GUI. At the least, most users will experience a bit of a learning curve as it's not immediately obvious how some aspects of the user interface work.



Nemo

8/10

Nemo implements the basics we've come to expect, with an emphasis on keeping it simple. However, it doesn't feel as compromised as *Thunar* as it offers a bit more. Beneath the sidebar there are icons that switch between Places (bookmarked folders and other locations) and a tree view. The bar along the top uses the breadcrumbs approach, but there is an icon to switch it to a URI display that can be directly edited and cut and pasted. Unlike *Dolphin*, the top

In the toolbar, three icons switch between the three viewing modes, and there is a search icon. There is a slider to control the zoom level of the main view. Of the traditional file managers we've looked at, *Nemo* strikes the best balance of simplicity without omitting desirable features with its default layout, even though *Dolphin* and *Krusader* can be coaxed into offering a lot more.

menu is enabled by default.

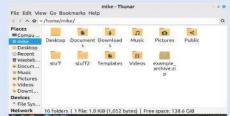


Thunar

7/10

The main *Thunar* window contains the files and folders, and there is a sidebar that's switchable between shortcut folders and a tree view. Along the top of the window, we find a box containing the folder URI, which is editable and pastable. In front of this, we have the back, forward, up and home buttons, and a search icon, as one might expect, but the fact that the toolbar can't easily be edited means that it's difficult to extend the user interface.

There are a few niceties that have been omitted from the layout. There are no buttons to change between different file display modes, and this has to be done within the main menu or via keyboard shortcuts. Similarly, there is no visible indication of current zoom level, although mouse wheel+Ctrl can adjust the zoom, and it's possible to go a bit larger than most file managers. All in all, it's a competent implementation of most of the commonly used features.



Simple simplification

If you want a basic file manager GUI, can it be achieved?

olphin is one of the most extensive and customisable file managers available for Linux, and starts off with an extremely conservative layout that we can't mark down for being too complicated. If you don't change anything, it's on the lower end of the scale in terms of interface complexity.

Nemo is a fork of the official Gnome file manager and attempts to retain some of the complexity that is constantly being taken out by the Gnome developers. The end result is a user interface experience that strikes the right balance between being slick without feeling dumbed down.

From the start, *Krusader* throws its multi-pane layout and fully populated menus and toolbars in the user's face. There isn't a great deal that can be done to reduce the number of interface elements that this technological monster offers. We can't see *Krusader* becoming the default Ubuntu file manager any time, and we can't see the *Krusader* users or developers wanting it to.

On the downside, *Thunar* doesn't offer a great deal of scope for ramping up the features and adding a lot of extra user

interface elements – however, it does well in this section because it presents the basics of what most users expect from a file manager. It even has several options that can be switched off if needed.

Most users can learn how to use it, but there isn't much hope for radically simplifying the user interface of *Midnight Commander*. It has a certain retro charm mixed with a pragmatic efficiency, thanks to keyboard-operated text-mode goodness, but it takes some time to learn, and it could never be considered a beginner's tool.

VERDICT

100			
DOLPHIN	8/10	NEMO	9/10
KRUSADER	4/10	THUNAR	8/10
MIDNIGHT COMMANDER	5/10		

Nemo is the most non-expert-friendly file manager here, with Dolphin and Thunar not too far behind.

www.techradar.com/pro/linux February 2024 LXF311 | 29

Extra tools

Beyond the basics, what special features are on offer?

rusader has a built-in folder sync tool, file comparison features, a mounted volume manager and features for working with system files. Generally, all of the menus have a little bit extra when it comes to features, and these extras could represent quite a lot of saved work because they are already configured and ready to go, even if some require installation of extra tools to the system to make them work.

Krusader, Thunar, Dolphin and Nemo all have good browser facilities for navigating and mounting network resources, to the extent that we didn't see a need to compare them with each other. The facilities of Midnight Commander in this area aren't quite as smooth. Having said that, Midnight Commander has a unique advantage when it comes to network use as it works extremely well over SSH in a terminal. This means that you can run it on another machine such as a Raspberry Pi for remote file management. Generally speaking, Midnight Commander offers a fluid workflow when combined with command-line work.

Thunar has a handy file search facility that searches from the current directory. Nemo takes things a step further because it can search within files. Dolphin can search by filename or content, too, but it can also link directly to other more advanced search tools. This requires some configuration and installation



Krusader's built-in search facilities are extensive, typical of the extra level of detail in every aspect of Krusader, but some might find the complexity overwhelming.

work, though. *Krusader* lacks these facilities but the built-in file search tools are more advanced out of the box because you can search by many criteria. Outside of search, *Dolphin* is aware of many external tools in areas such as disk space management and partitioning.

VERDICT								
DOLPHIN	9/10	NEMO	6/10					
KRUSADER	9/10	THUNAR	6/10					
MIDNIGHT COMMANDER	6/10							

Krusader has a lot of built-in facilities and pre-configured extras, while Dolphin has plenty of support for external tools.

Panels and views

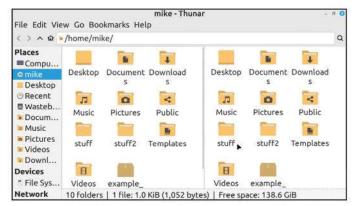
Clean views and flexibility are what we are looking for here.

he default *Dolphin* toolbar includes a button to split the main window in two, making it dual-pane when you need it to be. Middle-clicking opens a folder with its own tab. A left-hand panel contains the Places icons. Rather than this being switchable to a tree view, the tree view panel occupies the same area, which is vertically split if both are activated. On the right, an information panel can be activated. This gives a larger preview of the selected file along with detailed information. Finally, the Terminal panel can be activated, although it prompts you to install *Konsole*, the KDE terminal app, before it can be used. Once in place, this follows directory changes, and directory changes in the Terminal panel are reflected in the main window. Running all panels at once is possible, and all panels can be resized and torn off to be rearranged around the main window.

Nemo's side panel can be switched between Places and tree views. A second main panel can be activated for dual-pane operation, but there doesn't seem to be a way of activating that with an icon. There is also an optional command-line panel.

As ever, *Thunar* does a good job of offering all the expected features. The left-hand panel is switchable between Places, tree view and an image preview panel, but they can't all be activated at the same time. Middle-click can open a folder in a new tab rather than a separate window, but this is disabled by default.

Like a great deal of the *Midnight Commander* design, its use of panel layout is pretty much locked into a particular way of



Thunar goes a long way beyond just offering the file management basics. Here, we've switched on the split view, handy for some file-copying situations.

working. There are a few options, such as being able to split the panes vertically rather than horizontally, and it has a command-line pane at the bottom.

Lastly, Krusader starts off as a two-panel file manager, but each panel can be split again. This sub-panel can display a tree view, a file preview or a disk usage graph. Like Midnight Commander and Dolphin, a command-line terminal window can also be activated.

VERDICT								
DOLPHIN	9/10	NEMO	7/10					
KRUSADER	8/10	THUNAR	6/10					
MIDNIGHT COMMANDER	6/10							

Dolphin can add quite a lot of extras, and there's no downside to having them in reserve if you happen not to need them.

30 LXF311 February 2024 www.linuxformat.com

The verdict

File managers

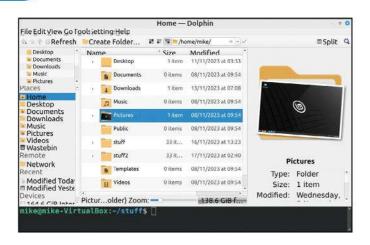
e've chosen *Dolphin* as the overall winner. The developers have decided to reduce the user interface complexity by default. With a bit of work, features can be re-enabled, meaning that *Dolphin* can scale all the way from a simple, lightweight file manager to a tool that will satisfy the most technical of Linux users. The built-in facilities are comprehensive, but *Dolphin* has a lot of support for external tools such as more advanced disk management and file search tools. In file management terms, there's not much that *Dolphin* can't do.

Krusader won't be everyone's cup of tea. However, it offers many of the advantages of a GUI file manager combined with a lot of the flavour and utility of a traditional text-mode file manager. It's not a beginner's tool, and nor does it try to be. As well as being a power user's paradise, it's incredibly configurable.

Thunar is a great file manager that does everything that it sets out to do without breaking the bank when it comes to resource usage. For many users, it will be everything that they will ever need in a file manager. There are limits to its configuration potential, though, and some more technically orientated users might miss the more advanced features of the other file managers.

Like its parent project, *Nautilus/Files, Nemo* has a minimalist look that just looks right on a business desktop. It can do most of what an average user requires of a file manager. It even packs in some slightly more advanced features, such as the dual-panel mode and multiple tabs. However, there is a limit to its expansion and customisation functions, and a more advanced user might find it constrictive.

We can't fault *Midnight Commander* for being a great implementation of a traditional text-mode file manager. Once you've learned the keyboard shortcuts and the main menu, it's a fast and accurate system for managing files. When used in repetitive file-copying situations, it's difficult to beat for efficiency. As an added bonus, it can even be used over SSH. However, GUI file managers evolved for a reason, and for casual use, it can feel like a lot of work to use. It also lacks visual refinements such as file previews.



1st

Dolphin

9/10

Web: https://apps.kde.org/dolphin

Licence: GPL 2.0 or later Version: 23.08.3

Smooth and efficient in use. Can be kept simple or expanded.

2nd

Krusader

8/10

Web: https://krusader.org

Licence: GPL 2.0 or later Version: 2.8.0

Powerful but complex. Highly accurate and applicable to repetitive file tasks.

3rd

Nemo

7/10

Web: https://github.com/linuxmint/nemo Licence: GPL 2.0 or later Version: 5.8.5

A standard tool that sits well on a business desktop.

4th

Thunar

7/10

Web: http://docs.xfce.org/xfce/thunar/start Licence: GPL 2.0 or later Version: 4.18.8

Covers the basic requirements and comes with a small memory footprint.

5th

Midnight Commander

7/10

Web: https://midnight-commander.org Licence: GPL 3.0 or later Version: 4.8.30

Technical users will love the efficiency. It mixes well with the command line.

>> ALSO CONSIDER

ROX-Filer is based on the file manager of Acorn's ARM-powered Archimedes workstations of the '80s and '90s. It has a traditional, minimalist layout and unsurprisingly reminds us of the file management tools of yesteryear. It's the default file manager of the Puppy-derived distros, such as EasyOS.

PCManFM is, like Thunar, a lightweight file manager that offers all the basics. We chose to look at Thunar instead because PCManFM removes a few extra features in favour

of lower resource usage. It's still worth looking at if you need basic facilities in a familiar-looking package when CPU and memory resources are extremely tight.

ZzzFM looks like a standard GUI file manager but works slightly differently. It shares some history with PCManFM and the no-longer-maintained SpaceFM file managers. It's the default file manager in the AntiX Linux distribution and is often combined with the Enlightenment file manager.

www.techradar.com/pro/linux February 2024 LXF311 | 31

STAY SAFE ONLINE!

Eat unwanted online nasties by sending packets to the Pi-hole. **David Rutland** helps you protect your browsing data with the ultimate in a paranoid's delight!



dverts are a plague on the internet – although we'll grant that in their current form, they're less intrusive than in days of yore. It's easy to look back on the late 1990s and early 2000s web through the haze of rosetinted glasses and view it as a utopia of high-quality independent blogs, free websites for everyone, and an explosion of experimental design along with non-predatory social media.

Lay your spectacles aside for a moment and consider that most of what you recall is wrong.

The internet at the turn of the century was never a utopia. Yes, you could get a free site from GeoCities, but you can make a better one today with GitHub Pages. Visually, the web was horrific, and adverts

were both gaudy and omnipresent. Pop-up ads and pop-under ads would obstruct your browsing and slow your computer to a crawl. Webmasters (as they were then known) would do anything to make a buck.

Google did a great deal to clean up the web and the advertising space in general. When was the last time your PC crashed due to an infinite stack of adverts appearing out of sight under your active window? Do you recall the whack-a-mole of fighting against endless cascade of pop-ups appearing in random locations, and hammering the big red X, only to discover that it was a trick, and you've actually launched a new wave of attacks?

That doesn't happen any more, and it's largely down to Google's dominance in the advertising space.

Even without running any kind of ad blocker, your browsing experience is much more pleasant and less brazenly predatory than it was two and a half decades ago.

There are trade-offs, of course, and the biggest is privacy. While websites no longer assault your visual cortex with an epilepsy-inducing assault, they instead quietly note what you're looking at. They measure how long you're on a web page and how you interact with it. Fair enough, you may say. Website owners need to know which articles are doing well and what people like to read. They'd be shouting into the void otherwise.

But consider how many websites you visit in a day. Your metrics, engagements and interests are noted down on each one, and tied together into a profile, which if it fell into the wrongs hands, probably wouldn't paint you in the best possible light.

The detail of a profile constructed from search and browsing data is alarming, and tracking companies can and do collect data on (deep breath) your age, location, devices, sex, sexual orientation, religion, income, contraceptive use, fertility, political views, race, health, your friends, your business dealings, how often you call your mum, and much, much more.

It doesn't stop when you're not using your computer either – thanks to that ever-present lump of metal, plastic and glass in your pocket. As well as the usual players, third-party companies pay impoverished developers to slip SDKs into their apps. If given appropriate permissions, these SDKs use Wi-Fi signal strength to map your location to within inches and beam that info back to the mothership.

All of this means that you get ads that are tailored to your interests – it's mildly irritating and gives you a mild sense of paranoia that you're being stalked, but at least you know about the latest Black Friday deals.

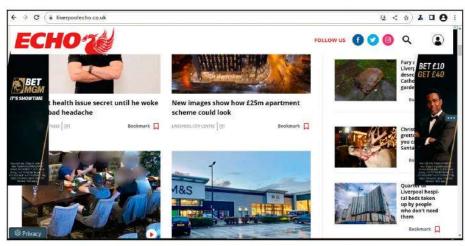
But consider what that intensely personal information could be used for in the hands of someone who wanted to do more than shill toasters.

There are many things that are illegal now that didn't used to be, and some texts that were popular in the 1990s are now criminal to possess.

In parts of the United States, women's search histories are being used as prosecution evidence in reproductive health cases. In November 2023, Russia outlawed the "international LGBT public movement" as



It may not look like much, but Pi-hole does more than just keep adverts out of your browser. It protects your privacy, too.



an extremist organisation, meaning that gay people could be treated as terrorists (always a handy go-to).

Rather than relying on search history, an 'advertising profile' would be far more useful in rooting out potential law-breakers and bringing them to justice. And God forbid that a malevolent actor could use your profile data to influence elections or referenda.

Ad blockers are a great solution, and this writer employs the *UBlock Origin* add-on for *Firefox* on his two laptops and two mobile phones. But there is a total of five 'proper' computers in the household, plus an additional three phones, a couple of Chromebooks, a streaming TV stick, an iPad, a few Raspberry Pis, and many other connected devices.

It's a pain to set up ad-blocking extensions on every device, and on some – such as the Chromebooks and Roku stick – it isn't even possible.

Pi-hole is a deceptively simple app designed to detect and thwart adverts and trackers on every device on your network – granting you peace of mind and the privacy to get on with your life without adding to a mysterious, privately-owned dossier of your activities. It can do a lot more besides. Read on to find out more...

Never having visited this site without Pi-hole before, it's odd that Reach considers this writer the type to bet on sporting fixtures.

» ETHICAL DILEMMAS

Making money on the internet is a difficult business. After all, if you can get something for free, why would you pay for it? Almost no one pays for news or articles any more – you can visit any website and get valuable written material for free. Or if you're part of the Google ecosystem, you don't even need to visit a website. Google presents interesting articles to you as part of the Discover experience.

Adverts make money for publishers because somewhere down the line, someone else is making money from a transaction and passing a share back to the site you're visiting. It could be Amazon, or it could be a bricks-and-mortar vape store in your home town that's looking to get into online sales and home delivery.

There are a few other business models available, and online ads is one of the less awful ones. You don't want to take out subscriptions for the dozens of publications you skim on a daily basis, and you don't want to have to create an account and log in, either. One other way sites make money is with affiliate links. This has made the internet a worse place, with otherwise respectable outlets spewing endless SEO-optimised 'Best-of' lists – essentially valueless content.

If you can come up with an ethical way publishers can make money from quality written content, please let us know.

CREDIT: Magictorch

Digging a fresh Pi-hole

Dig out that unused Pi and fire up the interweb to get started!



hen you visit a website on your computer (a word we're using to include any and all internet-connected devices), you type in the domain name. Your machine then consults a domain name service (DNS) to find the IP address of the site you want to visit.

If you're looking for back issues of your favourite monthly magazine, for example, you type https://linuxformat.com into the address bar, your DNS quickly consults its records, and directs the connection to our IP address, which is currently 18.134.167.236.

"Can I have index.html?" asks your computer.

"Here you go!" says the **LXF** server, returning a bare HTML document from its home on the AWS EU-West server farm. An HTML document isn't much good on its own, but it references other assets it needs to function and display properly. In the HTML head, your browser sees reference to a stylesheet containing

Pi-hole works by standing between your computer and the internet, and checking domain name requests against a list of known ad servers or tracking domains.

If the requested asset is on a blacklisted domain, *Pi-hole* returns an IP address of **0.0.0.0**. This address is effectively a black hole, from which no assets can be returned. The ad or tracking script isn't loaded.

If the asset isn't on a blacklisted domain, the request is forwarded to your usual DNS server.

This is almost exactly how a browser or host-based ad blocker works, with one key difference: *Pi-hole* takes over ad-blocking duties for your entire network. All of your PCs, phones, kettles, toasters, TVs and smart fridges are protected.

Fill your Pi-hole!

For very sensible reasons (see boxout, opposite page), we're advising that you use an actual Raspberry Pi to run *Pi-hole*. The model doesn't matter – although we recommend connecting over a wired connection (such as Ethernet) rather than Wi-Fi.

Before you start, make sure you have the Raspberry Pi Imager tool on your desktop PC. Insert your microSD card or plug in your USB drive, and select a distro. We recommend Raspberry Pi OS Lite, because it comes with less unnecessary software. You're not going to be running the Pi as a desktop computer, so you don't need desktop software – or even a desktop.

Once you've flashed your boot media, stick it in the Raspberry Pi, and power up. Now open a terminal and connect to your Pi over SSH:

\$ ssh pi@your.pi.local.ip.address

Update and upgrade your system with:

\$ sudo apt update && sudo apt upgrade

Because your Pi will be acting as a server, you need to ensure it has a static local IP address. Double-check its current IP address with:

\$ hostname -I

This command returns a whole load of IP addresses – you only need the first one. You also need your nameserver addresses (take a note of the contents):

\$ cat /etc/resolv.conf

The last piece of information you need is the IP address of the router. To find this, check the sticker on the box. Ours is **192.168.1.1**.

Use Nano to edit your /etc/dhcpcd.conf file:

\$ sudo nano /etc/dhcpcd.conf

At the bottom of the file, enter the information you just gathered:

interface [Either eth0 or wlan0]
static_routers=[Your router IP address]
static domain_name_servers=[Nameserver addresses]
static ip_address=[Your Pi IP address]/24

Save the file with Ctrl+O then Ctrl+X, and reboot the Raspberry Pi:

\$ sudo reboot

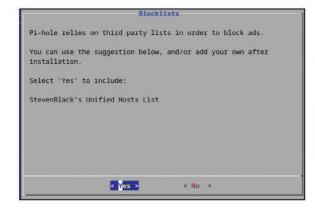
EVERYTHING BUT THE KITCHEN SINK

"Pi-hole takes over ad-blocking duties for your entire network. All of your PCs, phones, kettles, TVs and smart fridges are protected."

rules to make the page look pretty, and references to images, so you can examine what's on this month's cover. Your browser downloads all the ancillary parts, and puts them together as the **LFX** web devs intended.

Large sites tend to have their HTML pages stored on one server, and images and media files served from a subdomain or content distribution network (CDN). Each of these subdomains or CDNs has a different IP address, and your computer queries the DNS for each.

With a very few notable exceptions, adverts and tracking scripts are not served from the same domain as any of the other content you see in your browser.



If you don't install the StevenBlack blocklist, you'll have to add some manually later on. It's pretty comprehensive.



While we would love to completely block everything from Google.com, the rest of the family would riot and revolt.

The absolute easiest way to get *Pi-hole* on to your Raspberry Pi is with a script helpfully provided by the *Pi-hole* developers.

Install Curl with:

\$ sudo apt install curl

Then fetch and run the script:

\$ curl -sSL https://install.pi-hole.net | bash

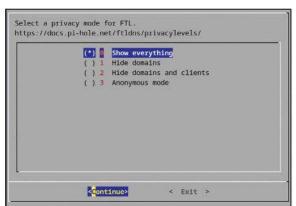
Enter your password when requested and let the *Pi-hole* automated installer script do its work. Hit OK when told that This installer will transform your device into a network-wide ad blocker! and OK again to acknowledge the call for donations, then Continue to acknowledge that your Pi has a static IP address.

At the next prompt, you need to choose an interface. If you're connected via Ethernet, choose eth0. Otherwise choose wlan0 for wireless.

You also need to choose an upstream DNS provider. Google is the default option, but we can't quite bring ourselves to trust the company that we're most trying to avoid. Cloudflare is a solid choice, and has historically resisted calls to block certain copyright-infringing websites.

The script then offers to install StevenBlack's unified hosts list. This is one of the most comprehensive, consolidated lists of advertising and tracking domains currently available. At the time of writing, it contains 144,626 entries, and while it doesn't necessarily catch all domains, it's pretty good, so go ahead and hit the Yes button.

You absolutely do want the admin web interface. Not only does it make administering *Pi-hole* a breeze, but it makes you look (and feel) like you're in command



With great power comes great responsibility. Pi-hole enables you to snoop on the web activity of all your users. You shouldn't, though.

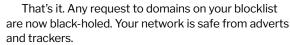
of a starship. You also want the lighttpd web server, and to enable query logging.

The privacy mode you select is up to you. If you select Show Everything, you can see all the traffic on your network, and which device it belongs to. This can be super-handy, as we'll show you on the following pages, but it's also ethically questionable.

Once you've answered the setup questions, installation completes, and you can visit the admin page in your browser at pi.hole/admin.

The last thing you may need to do is set your *Pi-hole* as the DHCP (dynamic host configuration protocol) server for your network.

Type your router IP address into your browser, and enter the password when requested. The DHCP section can be quite tricky to find. On our naff Vodafone Connect router (a Huawei in disguise), it was located in Settings > IPV4 > DHCP. Enter your Raspberry Pi's local IP address.



In the *Pi-hole* dashboard, you can see statistics on total queries, queries blocked, and the percentage of queries blocked. As shocking as these statistics are, after a few days, you'll want to head to the Query Log section to see what's going on in real time.

This section shows you every request from every device on your network as it happens. We have an old Microsoft tablet acting as a recipe platform and music player. That's all it does – so why is it sending regular requests and data to **finance.services.appex.bing. com?** It shouldn't be. If you find a suspicious entry like this, hit Blacklist to stop it from making contact again.





>> X86 PC-HOLE?

If you derive meaning from etymology, you might assume that *Pi-hole* is designed to run on a Raspberry Pi – a cheap single-board computer that arrived in the world in 2012. And you'd be right, but *Pi-hole* can run on anything with a network connection and a copy of your favourite Linux distro.

You can set up *Pi-hole* on the computer you're using now, on a screenless 2006 IBM ThinkPad, or on the RGB gaming monster down in the den. But unless you live alone and don't own a phone, you probably shouldn't – but you might do this for testing it out.

The idea behind having *Pi-hole* is that it protects your entire network from tracking and adverts. To do this, it needs to be powered on 24 hours a day, 365 days a year. If you shut it down to sleep, at best nothing on your network can connect to the internet; at worst, your browsing will be tracked by corporate ne'er-do-wells.

A modern laptop draws somewhere around 4 OW, and older or beefier tech pulls significantly more than that. Leaving your gaming rig powered on, fans humming 24/7, will send your electricity bills through the roof. A Pi of any variety costs less to run than any light bulb in your home. And they're cheap, too. *Pi-hole* can run on a Pi Zero, which at the time of writing will set you back less than a tenner. In our opinion, it's well worth the cost.



Beyond blocking adverts

Set up advanced blocking, plus keep control of your children's access.



or many people, the basics we've given on the previous pages are more than enough. Just install the software and let it do its thing. You can sit back and relax, knowing that you're ad-free on all the devices in your household.

But as we demonstrated with our kitchen tablet and its periodic connection to Bing(?), it's not always obvious that you're being monitored through your devices. Tracking networks don't always serve ads. Sometimes they just gather data. Ideally, you should leave *Pi-hole* active for a few weeks, and check the Query Log on a regular basis to see what's being sent from each device.

There's an element of experimentation here, and you need to be on the ball and ready to revert changes if they mess up your tech. Our Roku TV streaming media stick, for

instance, shows ads for upcoming movies and shows. They occupy significant screen real estate and also show up in the screensavers. *Pi-hole* reveals that Roku uses multiple subdomains.

Blacklisting one might kill streaming functionality altogether, while blacklisting another restricts you to a single DVD-style screensaver. It takes time to work out what's what. A couple of Roku domains that are safe to block and decrease your ad exposure are **scribe.logs. roku.com** and **cooper.logs.roku.com**.

Who blocks the blockers?

The default list installed with *Pi-hole* is very good and quite comprehensive, but you may find it doesn't suit your purpose. Maybe it throws up too many false

Output the last lines of the pihole.log file (live)

Automatic scrolling on update

Dec 8 17:36:43 dnsmasq[2393538]: forwarded youtube1.googleapis.com to 149.112.112.10
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 142.259.178.10
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 142.259.178.10
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 142.259.187.234
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 142.259.200.42
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 172.217.16.234
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 172.217.16.24
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 172.217.169.74
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 172.217.169.74
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 216.58.204.74
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 216.58.204.74
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 216.58.212.202
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 216.58.212.203
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 216.58.212.204
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 216.58.204.74
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 216.58.204.74
Dec 8 17:36:43 dnsmasq[2393538]: reply youtube1.googleapis.com is 21

If you want a real-time view of Pi-hole's inner workings without revisiting your terminal, you can check logs through the admin interface.

positives or it doesn't block some of the trackers you think it should. Fortunately, StevenBlack's list isn't the only blocklist out there.

Visit https://firebog.net for a list of lists. The page is divided into sections, making it easy to target exactly the type of content you don't want on your network.

For us, tracking is a more important problem than adverts, and we can scroll down to the Tracking & Telemetry Lists section to find a suitable candidate. You'll also find lists for malware, advertising and porn, and a lists of lists suspiciously titled Suspicious Lists.

While it may be tempting to add all the lists to *Pi-hole*, the risk of false positives affecting your browsing experience will shoot up, and you'll find it very hard to have a good time online. Instead, add more lists

>> KEEP CONTROL OF YOUR KIDS

The internet is harmful for children, we're told, and *Pi-hole* offers parents an intrusive insight into the sites they visit, when they visit them and on what devices. Monitoring your offspring's online activity without telling them is morally dubious, but imposing controls, and setting boundaries is not.

One way you can use *Pi-hole* is as a sort of timed access control for youngsters. The first thing to do is create three blocklists within your /var/www/html directory. We're calling them active_blocklist.txt, blank_list.txt and block_everything.txt. The first two files are empty, and the third contains only .*
- the regex to block everything.

In the *Pi-hole* interface, create a group for all of your child's devices.

Add the blocklist as http://localhost/active_blocklist.txt, and specify in the drop-down that it should only apply to your child's group.

Next, we're creating a couple of scripts in our **home** directory. The first, **block_script.sh**, used for blocking all sites, reads:

cp /var/www/html/block_everything.txt
/var/www/html/active_blocklist.txt
pihole -g

The second, unblock_script, reads: cp/var/www/html/blank_list.txt/var/ www/html/active_blocklist.txt pihole -g Next up, we create two new crontabs: \$ sudo crontab -e

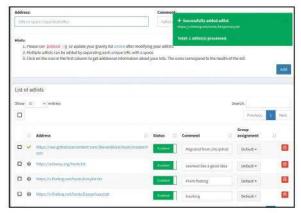
30 23 * bash /home/pi/block_script.sh 0 6 * bash /home/pi/unblock_script.sh

The first blocks all websites at 11.30pm, while the second unblocks them again at 6am.

You don't have to block everything, of course, but some regex relating to TikTok, Snapchat or whatever else the kids are using right now should ensure a good night's sleep.

It doesn't only have to apply to kids, either. If you're not getting much work done because you spend too much time on social media, you can easily block yourself during working hours.

36 **LXF311** February 2024



Adding blocklists to Pi-hole is simple – just take care not to add too many or stability may suffer.

as and when you feel you need them. The lists in green are the ones least likely to interfere with browsing, while still keeping ads and malware at bay.

Once you've decided on a list to add, right-click and copy the blocklist URL, and head back to the *Pi-hole* admin interface. Click on Group Management > Adlists, and paste the URL into the address box. It's handy to add a comment as well, because in the far future you'll have forgotten the reason why you chose this list to start with. Click Add to add the list.

Because Linux is all about puns, and because *Pi-hole* is based on the concept of a black hole for ads, you need to now click Tools > Update Gravity > Update. This lets *Pi-hole* know of yet more domain names to suck in. The wordplay continues as the output describes Neutrino emissions detected and Pulling blocklist sourcelist into range before informing you that, FTL is listening on Port 53. FTL stands for *Pi-hole*'s Faster Than Light engine, which manages DNS.

Regex gives you flexibility

The problem with blocking ads on a domain and subdomain basis is that advertisers, tracking companies and malware merchants can easily spin up a new subdomain in minutes. They do this regularly, and you should always keep *Pi-hole* and your blocklists updated to prevent it. But the blocklists aren't updated instantly with fresh threats, and it may be months before the maintainers twig on to a new ad source.

Fortunately, you can write your own rules using regular expressions (regex) – sequences of characters you can use for pattern matching within strings.

Most devices send information back to their manufacturers, developers or licence holders, for instance. Microsoft is terrible for this, and even Linux stalwarts aren't immune. Canonical does it, and so does Mozilla. They don't tend to hide the fact, though, and there's usually a setting to turn off telemetry in your favourite apps. But it's easier and quicker to create a custom blocklist containing regular expressions, rather than complete domain names, that you want to block.

Taking telemetry as an example, we could create a new blocklist text file with *Nano*:

\$ nano myblocklist.txt

Adding the word 'telemetry' on its own on a new line blocks all domains that contain the word 'telemetry'.

It's not ideal, though, and blocks domains where 'telemetry' is part of the URL. This would include the

Wikipedia page for telemetry, or a DDG search for the word 'telemetry'.

Using the anchor character – ^*\telemetry – blocks domains starting with the word 'telemetry', which is an improvement, but doesn't block domains such as incoming.telemetry.mozilla.org.

Regex can get very complicated very quickly, and the extensive *Pi-hole* documentation has some brief notes on it. For a more complete understanding, we suggest looking into the Mozilla developer regex cheatsheet.

After some time-consuming thought experiments, we determined that \b(?:\w+\.)*telemetry\.\w+\b will match instances where 'telemetry' is a subdomain, such as sub.telemetry.example.com or telemetry. subdomain.org. You can do this for patterns such as 'logs', 'tracking' or any other term that crops up regularly as a subdomain.

More than adverts

The world is full of scammers who want to take your money and buy themselves a new house on some Caribbean beach. One of the most common ways of doing this is to trick you into entering your bank details into a fake web portal. These can be hard to spot, as is evidenced on a regular basis by news stories of how vulnerable people lost their entire life savings while trying to buy new shoes on Facebook Marketplace.

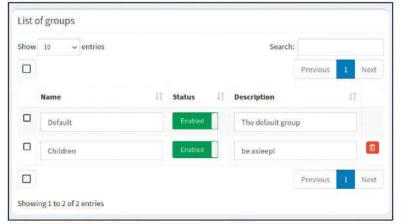
The domain names look close enough to your bank's domain to fool elderly relatives, thanks to character substitutions, particularly when punycode is employed.

Punycode attacks use domains that employ foreign characters that are rendered into something visually similar to English. You might think that clicking on a link for **natwest.com** will take you to the website of a popular UK bank. It won't, because the second character is actually a Cyrillic 'a'. In reality, the domain name is **xn--ntwest-3nf.com**, but it renders to look like **natwest.com** in your browser.

It's a good idea to create a blacklist to specifically thwart this kind of attack by excluding non-Latin characters that could be used to fool family members. Just add each character to a new line – you don't need to include obviously foreign characters like '3' or '4', just ones that look like they could fool your nan.

Scammers also tend to rely on free domains, and it was recently estimated that around 50% of scam sites are hosted on .tk TLDs – a domain belonging to the tiny Pacific island nation Tokelau. We recommend blocking any TK domains, along with ML, GA, CF and GQ.

Adding users to Pi-hole groups enables you to control their online activities as well as setting and enforcing bedtimes.









Taking your hole mobile

Protecting your browsing goes beyond just blocking malware locally.



our home is your castle. We get that. You pull up the drawbridge, bar the windows, and keep you and yours safe from internet threats. And you can stay safe in your castle as long as you like boarded up like a hermit from fear of what lies outside your walls. But at some point, you or a family member will need to leave the bunker and venture into the wasteland beyond the gates.

Maybe working from home isn't an option any more; it could be that your plan to home-school the kids overlooked your own unenviable academic record; perhaps you need to explore the grocery aisles of your local Lidl in search of something to prevent the inevitable onset of scurvy and rickets.

As you cross the moat and set your feet on the public pavement, a glance at the signal bars on your phone shows a Wi-Fi signal strength of two, then one, then zero. Your phone latches on to the nearest mobile were connected to your own private network, even when you're using mobile data, or sipping the free Wi-Fi along with your pint in 'Spoons. By passing all of your data through a VPN on your

affiliate websites telling you how to get the fastest

speeds, stream games, or watch foreign TV. These are

At their core, VPNs are a technology to create a

They enable you to access the internet as though you

secure and encrypted connection over the internet.

disingenuous attempts to skirt legality by disguising

their true purpose - which is, obviously, piracy.

Raspberry Pi, you're masking your true location, ensuring all your data is encrypted, and you're also enjoying the benefits of ad- and-tracker free browsing as afforded you by Pi-hole.

Setting up a VPN at home can be difficult and timeconsuming, and we've previously devoted entire features to it within this magazine.

PiVPN, however, is designed to be as simple and easy to install as possible, and as its name suggests, was built with the Raspberry Pi in mind.

At its core, PiVPN is a set of scripts used to install and set up both WireGuard and OpenVPN.

While the PiVPN project offers a variety of methods you can use to install the software, we feel it's in the spirit of things to use the neat one-liner. So, open a terminal and enter:

\$ curl -L https://install.pivpn.io | bash

The script makes sure any installed packages are up to date, and installs any dependencies you don't already have on your system.

You can click through most of the screens, but pay attention to the one titled DHCP Reservation. You should have already made sure that your Raspberry Pi has a static IP address, and clicking the default No option gets PiVPN to try to set it up again. Things can get complicated, and comments on the process in the script reveal: Not really robust and correct, we should actually check for dhcpcd, not the distro, but works on Raspbian and Debian. It doesn't exactly fill us with confidence. The user should be the one you're logged in as.

Later on in the setup process, the script asks you to make the choice between using WireGuard and OpenVPN. OpenVPN is older, more configurable, flexible and has arguably better documentation and compatibility, while WireGuard is its younger, hipper rival, boasting minimalistic design, easy configuration and lower latency.

From a purely pragmatic point of view, we recommend WireGuard because it uses less power, which means your phone battery will last longer.

Accept the default WireGuard port of 51820 and, most importantly, when PiVPN detects that you have Pi-hole installed, choose Yes when asked whether you want to use it as the DNS server for the VPN.

WATCH YOUR BACK

"You're convinced that various exploitative tracking networks are watching your every move. In this you are, of course, correct."

> mast, and you're no longer attached to your own network and no longer under the protection of Pi-hole.

You get ads as you browse the news at the bus stop, and you're convinced that various exploitative tracking networks are watching your every move. In this you are, of course, correct.

Away from home

Search online for virtual private networks (VPNs), and you'll see page after page of results for SEO-optimised

You should make sure the entire QR code is visible. **i@pi64:~ \$** pivpn -qr Client list David oneplus 8 pro lease enter the Index/Name of the Client to show: 1 Showing client David oneplus 8 pro below



We use a comically low resolution

to make sure

screenshots are

visible in the mag.

PiVPN checks your public IP address, so take a note of it before you click through.

After a few minutes, the *PiVPN* installation completes – and dishes up yet another subtle *Star Trek* quote in the corner of your screen – and you're returned to the command line.

You need to create client profiles for each of your devices, so enter the following command to start the process:

\$ pivpn add

PiVPN prompts you to enter a name for the client. Make this as descriptive as possible, because in six months' time, you're not going to remember that device3 is your son's iPhone, and device5 is your nephew's Nintendo Switch.

As you'd expect, *PiVPN* generates encryption keys and updates the server. The new config is placed in a newly minted **configs** directory within your **home** directory. Take a minute to email this to yourself in case you need it while you're away from your network.

We also consider it worthwhile to set up a few spare device profiles for use in an emergency.

Mobile VPN

With almost everything now set up to connect your mobile devices to *PiVPN* and bring them under the protection of the ever-vigilant *Pi-hole*, you're going to need an app.

If you went for the sensible WireGuard option, open your app store of choice, then search for and install a client app for WireGuard. We went with WG Tunnel on FDroid. Other client apps are available.

Click the blue plus button, and you have the option of adding your config from a file, adding from a QR code, or creating one from scratch.

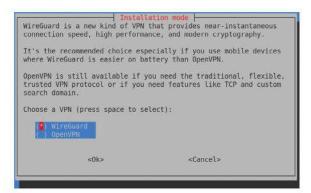
Because you're going to have to do this for every mobile device in your household, we recommend using the QR code option.

Back in the terminal, enter:

\$ pivpn -qr

From the list, select the device you're setting up. Point your phone at the computer screen, and wait.

You're returned to the Tunnels screen in the mobile app, with one single randomly named VPN tunnel next to a toggle switch. Simply flick the toggle switch to



Some diehards might insist on choosing OpenVPN over WireGuard for their PiVPN, but it will kill your phone battery.

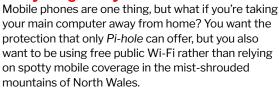
```
pi@pi64:~ $ pivpn add
Enter a Name for the Client: David_oneplus_8_pro
::: Client Keys generated
::: Client config generated
::: Updated server config
::: Updated hosts file for Pi-hole
::: WireGuard reloaded
::: Done! David_oneplus_8_pro.conf successfully created!
::: David_oneplus_8 pro.conf was copied to /home/pi/configs for easytransfer.
::: Please use this profile only on one device and create additional
::: profiles for other devices. You can also use pivpn -qr
::: to generate a QR Code you can scan with the mobile app.

pi@pi64:~ $ |
```

Two words in a single command is (almost) all it takes to ensure your devices are ad- and tracker-free when on the move.

protect your traffic from snoopers, then go out and do the shopping.

Everything, everywhere, all at once



Setting up your computer to connect to your VPN, and by extension *Pi-hole*, is fairly easy, and there is a WireGuard client available in the default repositories of most distros. On Ubuntu and other Debian-derived distros, you can install it with:

\$ sudo apt install wireguard

On almost every other distro we checked, the package is called **wireguard-tools**.

Use Nano to create a new config file:

\$ sudo nano /etc/wireguard/wg0.conf

Copy the contents of a *PiVPN*-generated config file into it, then start the connection with:

\$ sudo wq-quick up wq0

Now you're protected everywhere you go.

>> BELT AND BRACES

We love *Pi-hole*, and we love *PiVPN* almost as much. But it's a mistake to rely on these technologies entirely. Things can go wrong – especially if you use your Pi to self-host other sites or services.

One major potential drawback is that you can't access the internet from anywhere on your network if your Pi is down for any reason. Need to reboot or unplug drives? That's tough – there's no longer any DNS resolution, and everyone has an internet outage for a few minutes. Unless you hand back DNS responsibilities to your router.

False positives are another major issue, and you're going to spend a decent amount of time in the first few months working out why certain sites simply don't work at all. Other times, you may be relying on blocklists that are old and obsolete.

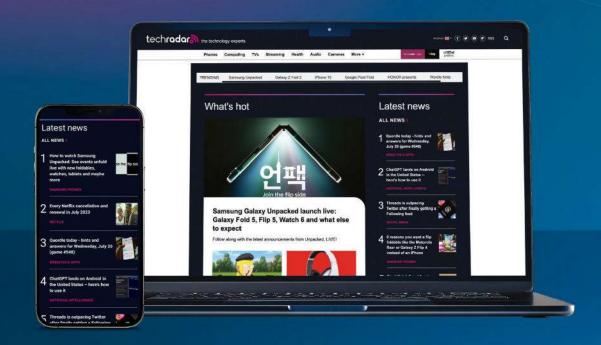
Because of this, we recommend using an on-device ad blocker in addition to *Pi-hole* – where it's possible to do so. In our experience, hardened *Firefox* combined with the excellent *UBlock Origin* has proved to be almost flawless.

Neither *UBlock Origin* nor *Pi-hole* will catch absolutely everything, but using them together, you stand a much better chance of keeping your private data private, and out of the clutches of malware mongers and greedy surveillance capitalists.





Meet the technology experts



- The world's most comprehensive technology website
- An unrivalled mix of news, opinions, reviews and features
- All-new design, new homepage, new features and special reports
 - Backed by over 300 years of editorial experience



www.techradar.com

Coolest Projects 2024 remote and in person

Start your Raspberry Pis! The coolest digital creators competition is ready to open in 2024.

o you like computers? Do you like the Raspberry Pi? Do you like winning stuff? Then you'll love Coolest Projects 2024. Registration opens on 14th February 2024 and closes on 22nd May, and the event gives young creators up to the age of 18 the chance to showcase their creations, finished or not.

There's no fee, it's open to everyone under 18 around the globe and for teams of up to five members. The global event will be live streamed, while in-person events will be held in the UK and Ireland. Registration for the Ireland event opens on 31st January 2024 and will take place on 13th April 2024.

It doesn't even matter if you haven't started your project. Coolest Projects is all about encouraging people to participate,



I Take part in the most exciting makers event this year!

share experiences and try out their ideas. There's loads of online learning materials to help you get started bringing your project to life. Find out more: https://online.coolestprojects.org

Les Pounder
works with groups
such as the
Raspberry Pi
Foundation to help
boost people's
maker skills.

> HATS OFF TO PINEBERRY PI

The Raspberry Pi 5 introduced a PCIe connector to the form factor and we wait with bated breath for the release of an official M.2 SSD HAT board. But it seems that Raspberry Pi Ltd has been beaten to market by Pineberry Pi's HatDrives. Created by Michał Gapiński and Mirosław Folejewski, there are two HatDrive boards: Top and Bottom. The Top board uses the true HAT form factor and is placed on top of the Pi 5. The form factor limits the drive selection to those that use 2230 and 2242, but they can be bought for little money. The Bottom board can be used with 2230, 2242 and 2280 drives, with 2280 drives being cheaper and plentiful thanks to their wide adoption in desktops and laptops.

Both boards work with PCIe x1 Gen 3 standards, offering plenty of performance for the Raspberry Pi 5. These boards cannot be used with older Raspberry Pis. For those boards, a USB-to-SATA/NVMe adaptor would provide the best and most cost-effective upgrade route.

With PCle-based SSDs now becoming available for the Raspberry Pi, we must consider that the Pi 5 could become more than a single-board computer for makers. It could become the first Linux computer for many users. In the past it would have been an old laptop, but now we have the opportunity for great hardware performance and low power consumption at a low price.

HatDrives start at €20 for the Top and €26 for the Bottom. Learn more and buy a board from https://pineberrypi.com.

No M.2 4 U

Lack of Pi space.

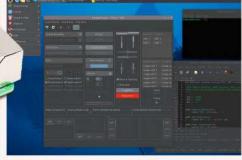
The announcement that there is an M.2 PCIe adaptor coming to the Raspberry Pi 5 in the form of an add-on HAT is very welcome news, but the lack of an embedded M.2 slot on the Pi 5 itself was something of a disappointment. However, a comprehensive blog post outlines why it simply could not be. Read the explanation at:



Darkness arrives!

In time for Christmas!

We've been using dark themes since before they were cool – our goth leanings are strong – so it's great news that finally the Raspberry Pi OS is getting a dark theme. Our late-night eyes will be ever thankful. Apparently, Eben wandered past the Pi OS devs and suggested a dark theme would be nice. Next thing you know, it's done. You can find out more at: https://bit.ly/lxf311dark



Our teen selves would love it.

Ubuntu 23.10 for Pi

The ever dapper **Les Pounder** has been using Ubuntu since Dapper Drake and has been looking forward to sampling this Pi-based iteration.

IN BRIEF

This is the latest Ubuntu OS for the Raspberry Pi 4 and the newly released Raspberry Pi 5. It provides a slick desktop experience on the Pi 5, but it's a little sluggish on the Pi 4. GPIO access is there, but your HATs won't work - Ubuntu is not alone in that issue, however. If you like the Ubuntu OS, but need it on a Raspberry Pi, look no further.

buntu 23.10 for the Raspberry Pi sees the desktop OS going toe to toe with the official Raspberry Pi OS. But it seems that this time, Ubuntu has the winning combination. Ubuntu 23.10 is for Raspberry Pis with 4GB or more RAM. This limits it to the Raspberry Pi 4, Compute Module 4 and the Pi 5. We tested it on a Pi 4 and Pi 5 with 8GB of RAM, running the CPU at stock speeds.

First impressions: this is a true desktop Ubuntu experience. The Gnome 45-based desktop booted in 24.84 seconds from microSD and it felt just like our laptop. On the Raspberry Pi 5, the desktop was very responsive; on the Raspberry Pi 4, it was

sluggish. The differences clearly being the faster CPU and microSD card access of the Pi 5.

One Pi 5-centric quirk is that the official Active Cooler fan is always at 100%. This means the fan is noisy. The Ubuntu team has been made aware of this.

We wanted to see if the OS was heavier than the official Raspberry Pi OS. A heavier OS makes the CPU of a low-power machine work harder and get hotter. Ubuntu 23.10 on the Pi 5 ran just as well as Raspberry Pi OS Bookworm. At idle, the CPU temperature on the Pi 5 was at 31.2°C, largely down to the fan always being on. Under a five-minute stress test, it reached 48.8°C. Nowhere near the 82°C thermal throttle of the Pi 5's CPU and well under the 59.3°C stress test for Raspberry Pi OS. Again, this is down to the aggressive active cooling fan. If we took the cooling off, it would thermal throttle, just like Raspberry Pi OS does.

Browsing with *Firefox* is relatively smooth. We loaded an ad-heavy page and saw the main content quickly load, while ads took only a few seconds longer.

Playing *Big Buck Bunny* at 1080p full-screen, after one minute we saw 706 from a total of 3,167 frames dropped – roughly 22% of the frames. This is mostly down to loading the 'stats for nerds' feature and going



Ubuntu 23.10 looks great on the Raspberry Pi 5, and the added power makes it snappy.

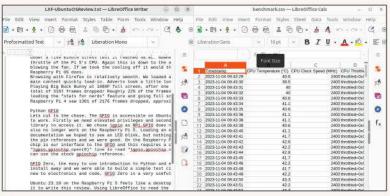
full-screen. The same test on a Raspberry Pi 4 saw 1,301 of 2,176 frames dropped – approximately 59%.

Let's cut to the chase. The GPIO is accessible on Ubuntu but your HATs are unlikely to work. Firstly we need elevated privileges, secondly we need a module or library to access it. We chose Igpio for Python as RPi.GPIO does not work, hence why many HATs also no longer work on the Pi 5. Loading an example from the Ubuntu documentation, we hoped to see an LED blink, but nothing. We checked the wiring and pin references – all good. On the Pi 5, the new RP1 chip is our interface to the GPIO and this requires a change to the Igpio.gpiochip_open(0) line to read Igpio.gpiochip_open(4). Older Pis can use the stock reference.

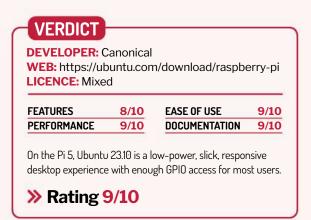
Ubuntu 23.10 on the Raspberry Pi 5 feels like a desktop computer. We happily used *LibreOffice* to write this review and to read the benchmark CSV file.

Will it pull us away from our Ryzen 5 5600X machine? No, but this is the closest that Ubuntu on the Raspberry Pi has felt like a desktop since its release.

We liked Ubuntu 23.10 on the Pi 5 – more than Raspberry Pi OS. The GPIO issue of Raspberry Pi OS brings the two OSes into parity, but Ubuntu just pips the official OS because it feels like a 'proper' OS.



Yes, we worked on this review using Ubuntu 23.10 on the Pi 5. It felt like we were using our laptop.



42 LXF311 February 2024 www.linuxformat.com

Bambu Lab A1 Mini

Good things come in small packages is something **Denise Bertacchi** never says – now she's eating her delicious words.

SPECS

Vol: 180x 180x180mm Type: PLA, PETG, TPU (single spool) (up to 300°C) **Extruder:** Direct drive Size: 0.4mm high flow **Platform: PEI** textured spring steel sheet. heated Levelling: Auto Sensors: Runout, odometer Comms: Wi-Fi, Bambu-bus, microSD **Control:** 2.4-inch colour touchscreen Footprint: 347x

314x365mm

Weight: 5.5kg

he A1 Mini is half the price of a stripped-down P1P Combo, yet packed with more features. It has an intuitive full-colour touchscreen, quickchange nozzles and an AMS (Automatic Material System) unit less prone to jamming.

Like its big brothers, the A1 Mini arrives 95% assembled, with a direct drive and steel-tipped full-metal hotend that can reach 300°C. Its top speed is 500mm/s but it has 'only' half the maximum acceleration, at 10,000mm/s²; still enough to make it the fastest Cartesian printer we've reviewed.

For a single spool, you push filament through the Bowden tube to the toolhead, then tap Load Filament on the menu. For

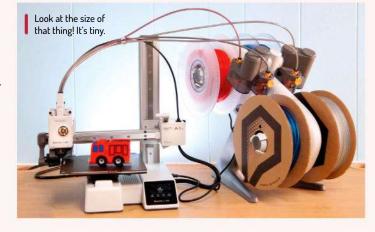
the AMS, you push spools on to the motorised spindles and poke the filament into the intake. The machine detects the filament and draws it in. The AMS has an RFID reader, which allows it to read tags on Bambu Labbranded filament so it can identify colour and type.

The nozzles contain the heat break and cooling fins, which like the Prusa MK4 nozzles, prevent leaks from haphazard user installation. Unlike previous Bambu Lab machines, these nozzles are a cinch to install.

The A1 Mini has a silicone brush near the back of the plate for wiping the nozzle before it taps the bed for calibration. It also has a weird tray at the end of the X gantry for depositing purged filament, which it then flings off to the side in a hilarious fashion.

There's a filament run-out sensor in the toolhead and an odometer in the AMS to detect broken filament and tangles. It also has a camera for monitoring and time lapses, but there is no AI programming to detect fails.

There are drawbacks to the AMS. The printer wastes material when it cleans the nozzle between colours. For our pair of fire engines, it took 166g of filament to print and it flushed 115g; otherwise it was 200g. The second drawback is loss of speed. There's a pause between layers to do a nozzle clean cycle. The fire engines took 22



hours and 25 minutes to print in four colours. As a single colour, it would only take four hours and 19 minutes.

Bambu Lab has its own custom slicer, which is a fork of PrusaSlicer. You need to use Bambu Slicer or the open source OrcaSlicer with any Bambu printer because it can't read plain gcode – it uses 3mf instead. The slicer is also the primary means of transferring files via the cloud or local LAN to the printer. Of course, you can also place files on the SD card and walk them to your printer.

The Benchy Speed Boat print (standard two walls, three top and bottom layers, 10% grid infill, 0.25mm layer height, 0.5mm layer width) finished blazingly fast at 19 minutes and 15 seconds. The layers are smooth, without any ringing or layer shifts. It gets a little sloppy on the overhangs and around the bow from lack of cooling, but considering this print should take 46 minutes with default settings, the print is fantastic.

The Bambu Lab A1 Mini is the fastest bed slinger we've clocked so far, and nearly as fast as Bambu's larger Core XY machines. It's much easier to use than Bambu's bigger printers but lacks the build volume. The AMS takes up more space than we'd like, but is so much easier to use and maintain, it's worth the counter space.



VERDICT

DEVELOPER: Bambu Lab **WEB:** http://uk.store.bambulab.com **PRICE:** £409 AMS combo (£269 printer only)

 FEATURES
 9/10
 EASE OF USE
 9/10

 PERFORMANCE
 9/10
 VALUE
 9/10

Bambu Lab's budget bed slinger is fast, quiet and easy to use. The build size is small, but roomy enough for many practical prints and fun models.

>> Rating 9/10

Pi PICO

Build a Pi Pico electronic piano

Les Pounder is trying his best to make his mum proud by not taking apart an electronic piano!



Les Pounder is associate editor at Tom's Hardware and a freelance maker for hire. He blogs about his adventures and projects at http://bigl.es.

s I write this, Christmas is on the horizon and it's got me thinking of my childhood. My mother bought me a small piezo-speaker piano in the hope that I would learn to play. That dream was never realised but I did learn to take it apart and see how it worked, and now we can do that using the Raspberry Pi Pico.

In this tutorial, we are going to use a Raspberry Pi Pico or Pico W running MicroPython to create a Pi-ano. We'll use the Solfège syllables, a music education method used in schools to teach pitch and basic music comprehension. You'll most likely

know them as do, re, mi, fa, sol, la, ti. We will create a piano interface using seven push buttons, and using a single buzzer and some clever coding, we will produce 'music' to 'delight' our families.

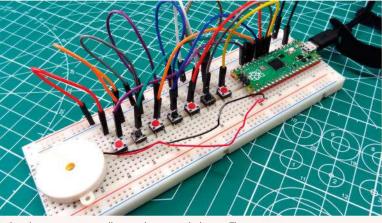
Our electronic piano is really seven buttons and a buzzer. The magic that makes it work is a little bit of MicroPython code. file and copy it to the root of the RPI-RP2 drive. This flashes the new firmware to the Pico.

Building the hardware

The hardware build is split into two sections. The inputs (buttons) and the output (buzzer). There are seven buttons connected to GPIO pins on one side of the Pico. One leg of the button is connected to the GPIO pin, and the other is connected to a GND rail. We need to connect the GND of the Pico to the GND rail so that we have a common GND (0 volts reference). The GPIO pins are pulled high to 3.3V in the code and when we press the button, we connect that pin to the GND rail, the trigger to play a tone. The buzzer is connected to GP16 and GND. Use a little Blu Tack or modelling clay to secure it to the breadboard. It also helps amplification.

Setting up Thonny

While holding the BOOTSEL button, connect your Raspberry Pi Pico to your computer. Go to www. raspberrypi.com/documentation/microcontrollers/ micropython.html#drag-and-drop-micropython and download the version of MicroPython for your Pico or Pico W. Open your file manager, go to the downloaded



to the board and we can start writing code. Coding the project

\$ sudo snap install thonny

In a new tab, we start the code. First we import two modules: machine, used to access the GPIO, and utime, which is used to add delays to the code:

Using your distro's package manager, install Thonny.

For the latest Ubuntu release, we use a Snap package:

Open Thonny and connect the Pico to your

Interpreter tab. Set the interpreter to MicroPython

(Raspberry Pi Pico) and set the Port to match the location of your Pico. Click OK. Thonny now connects

machine. Go to Tools > Options and select the

import machine import utime

Next we create two objects. The first is a list, used to store the GPIO pin reference for each button. The other is a variable to store the GPIO pin for the buzzer.

button_pins = [6, 7, 8, 9, 10, 11, 12] buzzer_pin = 16

Using a dictionary, we store the frequency for each Solfège syllable. The dictionary has a key-value structure and we use the name of the syllable as the key. By looking in a dictionary for that key, it returns the

YOU NEED

> Pi Pico or Pico W > A buzzer >15x M2M

jumper wires >7x push

buttons Large

breadboard Blu Tack, modelling clay, double-

sided tape > Code:

https:// github.com/ lesp/

LXF311-Pi-Ano/archive/ refs/heads/ main.zip

There are a lot of wires to this

circuit. Follow each

one carefully and

double-check your work. Even

we missed a

wire for sol.

value: the frequency. These frequencies are not perfect, so tweak them to your ear.

```
frequencies = {
    'DO': 261.63,
    'RE': 293.66,
    'MI': 329.63,
    'FA': 349.23,
    'SOL': 392.00,
    'LA': 440.00,
    'TI': 493.88,
}
```

Using a **for** loop, we create a list called **buttons** that sets up each of the buttons as an input, and pulls its corresponding GPIO pin up (**high**, **1**, **True**). GPIO pins can

be pulled up or down to GND. By pulling the pins up, we set their status to **HIGH**. Pressing the button connects that pin to GND and changes its state to **LOW** (**down**, **0**, **False**) and we use that as a trigger to play a tone.

buttons = [machine.Pin(pin, machine.Pin.IN, machine. Pin.PULL_UP) for pin in button_pins]

Set the buzzer GPIO pin to use PWM (pulse width modulation), enabling us to set the tone's frequency. buzzer = machine.PWM(machine.Pin(buzzer_pin))

Create a function, **play_tone**, which takes two arguments – the frequency of the tone and its duration: def play_tone(frequency, duration):

Inside the function, we set the duty cycle (how long the signal is active) to 50% and then set the buzzer frequency using an integer value. This is the value returned from the frequencies dictionary.

buzzer.duty_u16(512)

buzzer.freq(int(frequency))

Set the sleep time using the duration argument, then set the duty cycle to 0 to turn off the buzzer.

utime.sleep_ms(duration)

buzzer.duty_u16(0)

Now we move to the main loop. This uses a **while True** and a **for** loop to constantly check the status of the buttons.

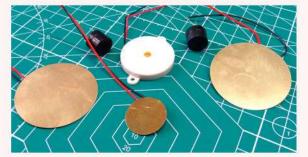
while True:

for i, button in enumerate(buttons):

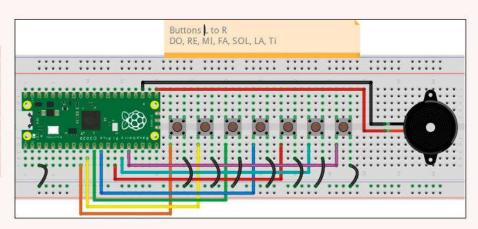
Inside the **for** loop is an **if** conditional test that checks for each button press. The default state of the GPIO pin for each button is **1** (**True**, **high**, **up**), but when we press the button it changes to **0** (**False**, **low**, **down**).

if button.value() == 0:

When the button is pressed a variable called **note** is created and in it we store the key (the Solfège syllable)



Buzzers come in all sorts of forms. Simple piezo discs, active buzzers (black) and an amplified piezo buzzer (white).



and then get the frequency value, storing it inside a variable called **frequency**.

note = list(frequencies.keys())[i]
frequency = frequencies[note]

Using a little string formatting, we print both the GPIO pin reference for the button press and the Solfège syllable:

print("Button {} ({}): Playing {}".format(i, note, note))

We then play the note using the **play_tone** function. We pass the frequency and duration to the variable. For shorter bursts of sound, reduce the duration (**500**). We found that 200 was a good choice. Lastly, outside of the **if** condition and **for** loop, but still inside the main loop, we add a short delay to prevent debounce. Debounce is when the loop is so quick that it thinks we pressed the button twice. Adding a very short delay reduces the chance of this happening.

play_tone(frequency, 500) utime.sleep(0.1)

Save the code to the Raspberry Pi Pico and **Pl-ano.** py, and click Run > Run Current Script (or click on the green play button, or press F5) to run the code. Now press the buttons and perform your own concerto.

» MAKING MUSIC WITH PICO

Making music with microcontrollers is not limited to mere beeps and boops. Even the humble Pi Pico can make rich pieces of music, with a little help from its friends. The MIDI (Musical Instrument Digital Interface) standard is a means in which a device, say a Pi Pico, can send instructions to a real instrument. The instruments are largely electronic, such as piano, synthesiser and drum machine, but we can send instructions from a Pico and hear the output on such devices.

In normal operation, there is a controller device, which leads the group and sends instructions to a drum machine. But with clever coding or a MIDI digital audio workstation, we can compose entire pieces of music and send the instructions to the machines.

CircuitPython, a fork of MicroPython from Adafruit, has a MIDI library that makes sending MIDI instructions as easy as Python code. All we need to do is tell the controller what note to hit, how long to hold it and when to release. Adafruit also has a MIDI FeatherWing (an add-on board for its Feather range of boards, including the RP2040), which enables direct connection to musical devices, no hacks required. More details can be found at https://learn.adafruit.com/adafruit-midi-featherwing/circuitpython-midi-example.

>> GET YOUR Pi FILLING HERE Subscribe now at http://bit.ly/LinuxFormat

QBITTORRENT

Credit: www.qbittorrent.org

Torrent your slice of Pi

Torrenting isn't all about taking – **Christian Cawley** helps you use a Raspberry Pi to serve open source ISOs on the BitTorrent P2P network.



Christian Cawley owns more Raspberry Pis than he has children. He spends equal time with them all, and his favourite is the small one that arrived in 2017.

ne of the key strengths of the Raspberry Pi is its fire-and-forgetability. That key quality where you can set instructions and let the machine simply perform the assigned tasks is incredibly useful; there is no chance of automatic restarts following unwanted updates, for example, or antivirus scans disrupting scheduled tasks.

Hook up some external USB storage and the Pi can become a data server (for example, a PC backup system, as in **LXF310**). A similar setup can be used in a different way: to serve ISOs to the BitTorrent network.

No, don't choke on your pastry; this is entirely legal if you stick to files that are intended to be shared.

Is BitTorrent still a thing?

For people who don't like paying for movies, music and video games, BitTorrent became popular around 15 years ago as an alternative to *LimeWire*, *eMule* and various other download managers. The focus was on copyrighted material, and the servers hosting the files were rightly taken down.

BitTorrent differed in method, which is why it is still used. No central server hosting copyrighted material can be taken down because none exists. Attempts to remove illegal hubs like the Pirate Bay have turned into a game of whack-a-mole (there are dozens of clones).

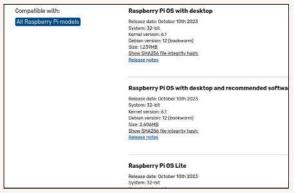
Because BitTorrent's peer-to-peer networking model is very efficient, it is widely used. The original decentralised internet technology, BitTorrent files are downloaded and uploaded to and from a network of computers, in whatever order is the most efficient.

Anyone who has heard of BitTorrent is no doubt aware of its illegal uses. But just as *Kodi* can be misused, so can BitTorrent. You don't want to misuse a Pi-based torrentbox, so what can be legally served?

All manner of legal torrents are available, some of which are listed on specialist sites. The one we're most interested in is LinuxTracker (https://bit.ly/LXF311-linuxtracker), which lists almost every Linux ISO that is available to download. LinuxTracker's collection of BitTorrent trackers lists pretty much all the distros you can think of. Its main page is split between recently added torrents and the most popular ones. You'll find everything from Linux Mint to CentOS, not to mention Raspberry Pi OS. So, let's make Raspberry Pi OS the first ISO you serve with your Raspberry Pi torrentbox.

Hardware options

This project should work reliably on a Raspberry Pi 2 and later. A more recent model will, of course, deliver



The Raspberry Pi site lists BitTorrent alongside direct download links for the OS. Depending on your connection, BitTorrent can be faster.

better results – we've used a Raspberry Pi 4 with 8GB of RAM. A key aim of this project is to build the torrentbox as a headless unit, so you don't need a monitor, keyboard or mouse connected to your Pi. If you prefer to work directly on your Pi rather than over SSH, then you can, of course, connect these devices.

Reliable torrenting is possible over Wi-Fi, but for the best results, a direct Ethernet cable connection from your router delivers more reliable speeds, up and down. Speaking of which, while torrenting is possible over any network connection, the faster your internet speed, the better. It can also take up quite a bit of bandwidth, depending on how many ISOs you're hosting. A key consequence of this is that it slows down the internet for other people in your property, so it's good to start with sharing just one or two ISOs.

You have the option of connecting a USB stick or USB hard drive to your Raspberry Pi to provide extra storage. This is preferable to relying on the microSD card, and reduces the load on the Pi.

Typical ISOs are 4GB. These can fit comfortably on most microSD cards, but if yours is a smaller capacity card, it would be wise to consider bringing an external storage device into the mix.

Best Linux torrent software

Three torrenting tools are known to work well on the Raspberry Pi: *Deluge, Transmission* and *qBittorrent*. But which one is best for this project?

Having tried them all, we've found that *qBittorrent* just edges it. All offer headless access with web-based interfaces, but *qBittorrent* is that bit easier to install and set up than the others. Meanwhile, the browser interface seems more responsive. If you're not sure,

QUICK TIP

As ever, you'll get the best results from your Raspberry Pi if you set this project up on a fresh installation of your favourite OS. We used Raspberry Pi OS Lite.

first try these steps with *qBittorrent*, then restart the project from scratch, trying out *Transmission* and *Deluge*, along with the steps each of these requires.

Set up the torrentbox

The Raspberry Pi torrentbox is best run as a headless server project, accessed over SSH. You can start with a fresh installation (for headless, select a Raspberry Pi Lite release in the *Raspberry Pi Imager*) or continue with your existing Pi operating system. If the former, enable SSH in the *Imager* tool; if the latter (or a desktop build), enable SSH in Raspberry Pi Configuration > Interfaces, and click OK to confirm.

Before you do anything else, update and upgrade your Raspberry Pi OS:

\$ sudo apt update && sudo apt upgrade -y

Then install the qBittorrent headless software:

\$ sudo apt install qbittorrent-nox/

The next step is to check the available commands:

\$ qbittorrent-nox --help

Before downloading anything, determine a directory for downloads. You're probably in the **home** folder (input pwd to check); ideally you want to be in Downloads or a subdirectory. We'll assume the latter:

\$ sudo mkdir /torrents

Next, it's worth taking a look at the various options available with this tool:

\$ qbittorrent-nox --help

For example, you can download a .torrent file, then make it available to anyone else who downloads it.

Launch the web interface

While it is possible to manage your torrents from the command line, it's simpler to enable the browser-based user interface. This is a feature in *qBittorrent* that can be enabled from the desktop app, but as you may be using a lite Raspberry Pi OS, this isn't an option.

Instead, you can enable the web UI from the terminal with this single command:

\$ qbittorrent-nox

Pay heed to the warnings, and make a note of the credentials. You'll see a notice to change the password, which you can do in Options > Web UI. Various other options are available here (more on these later).



Changing the password means opening the Options cog and clicking the Web Ul tab. Under Authentication, add a new password.

Going forward, if you require continued access to the terminal, you may decide to run *qBittorrent-nox* as a daemon. (This means it runs in the background and doesn't hijack the terminal window.) Do this with:

\$ qbittorrent-nox -d

We found that after cancelling the web UI in the terminal (with Ctrl+Z), we were unable to subsequently start it in daemon mode. Instead, we had to first reboot the Raspberry Pi and try again.

To start seeding a torrent from the web interface, begin by downloading or creating a torrent file. For example, you can download the .torrent file for any Raspberry Pi OS build from the website (https://bit.ly/LXF311-pidownloads) and save it to your Raspberry Pi. Then, click the blue plus icon (Add Torrent File) and in the resulting Upload Local Torrent window, click Browse to find the .torrent file.

Ensure the Save Files To Location field matches the location you set earlier, add a category if you need to, and click Upload Torrents. You may use the limit rate settings if you have concerns over bandwidth; it's best to avoid these until you get used to things, however.

At this point, *qBittorrent* begins downloading the ISO from the location. Once this is done, it becomes ready to share on the P2P torrent network. By hosting the file, you're making the Raspberry Pi OS available to anyone who needs it, supporting the Raspberry Pi

>> BITTORRENT TERMS YOU SHOULD KNOW

New to BitTorrent? Some of the terms may be confusing, so here's a quick primer.

Peer - a computer connected to the P2P network and equipped with BitTorrent software. This enables the sharing of data files between peers, ensuring a uniform distribution of data.

P2P – short for peer to peer, which refers to the network architecture used in BitTorrent. Every peer in a P2P network has privileges, although some may perform different tasks. For example, when sharing a file, different peers share different data.

Torrent – a file, denoted with a .torrent extension, it holds a hash, tracker list and file and folder details. They can be downloaded via a browser, but require a client to download the file's data,

such as Deluge, qBittorrent, and Transmission.

Tracker – a server that manages the communication between the computers in a P2P network. No file data is stored on the tracker; it is provided solely to ensure all peers can share data.

Seed – the aim of this tutorial is to encourage you to use your Raspberry Pi to share ISOs. As a seed, you have the complete file, seeding (sharing) it to the P2P network. Anyone who downloads the file then becomes a seed.

Leecher – the default position of anyone who opens a .torrent file in a client. Once it is loaded, data is shared from the previously leeching peer; another peer may need that data, to which it is now also shared.



>> MAGAZINES ARE OLD SCHOOL P2P Subscribe now at http://bit.ly/LinuxFormat

CREATE A TORRENT IN THE TERMINAL

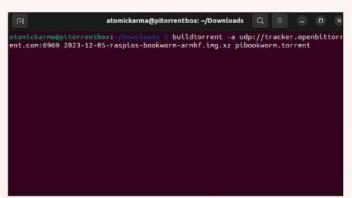


Install Buildtorrent
While flexible, qBittorrent doesn't have the necessary functionality to create a torrent from the command line.
Consequently, you need an extra piece of software. Arguably the easiest way to create a torrent from the terminal is to use Buildtorrent. You can install it with:

\$ sudo apt install buildtorrent



Find a tracker server
To create a brand new torrent, you need a tracker server
URL. Where do you find one? Various open trackers can be used.
You'll find a list of them online (https://bit.ly/LXF311-trackers)
but we're going to use the one provided by OpenBitTorrent:
udp://tracker.openbittorrent.com:6969

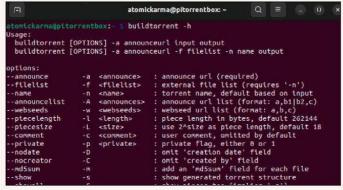


Create a torrent

Any file you have permission to use and share can be turned into a torrent. To illustrate, we're going to use the Raspberry Pi OS from the main text:

\$ buildtorrent -a udp://tracker.openbittorrent.com:6969 2023-12-05-raspios-bookworm-armhf.img.xz pibookworm. torrent

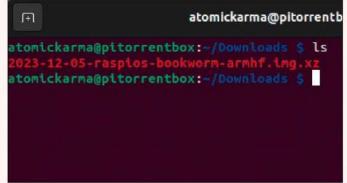
Wait while the torrent is created. Larger .torrent files take longer.



Check the Buildtorrent help

Using Buildtorrent requires you to input a straightforward command that announces the torrent's tracker and the data file you're sharing. The format of this command is:

\$ buildtorrent [OPTIONS] -a announceurl input output However, it is worth taking a moment to check the help file (buildtorrent -h) to view further options.



Check the ISO location
Before creating the torrent file, just take a moment to confirm the location of the ISO file you plan to share. This probably means browsing through directories using cd and ls, unless you already know the location and filename of the file for which you plan to create a torrent.



6 Share the torrent file With the torrent file created

With the torrent file created, it can now be shared online. While there are various ways you can do this, it seems appropriate that a Linux ISO torrent should be uploaded to LinuxTracker (https://bit.ly/LXF311-linuxtracker). Of course, duplicating torrents isn't particularly useful to anyone, so consider ways in which you can safely customise a Linux distribution before sharing it.

website by reducing its server overheads and generally being an awesome person. Don't forget, you can also create your own torrents to share on the P2P network (see walkthrough, left).

Understand the web interface

Like the desktop client, the *qBittorrent* web UI provides the tools to download (and upload) torrents. The File and Tools menus are across the top, the main pane highlights any .torrent added to the tool, and the left pane provides a quick status overview. The Filter box lets you quickly find what you're looking for. This is useful if you have a lot of torrents on the go.

The Options screen is worth understanding. Activated via the Tools menu or by clicking the cog, this is split into seven tabs: Downloads, Connection, Speed, BitTorrent, RSS, Web UI and Advanced.

The Downloads tab lets you control how torrents are added and saved. It gives you the tools to set email notifications and launch external applications upon download completion.

The Connection tab is there for tweaking protocols, ports and proxy servers, and setting connection limits and IP filters. On a related note, the Speed tab lets you set a global rate limit (individual download rate limits can also be set manually).

Some privacy controls can be set in the BitTorrent tab, such as enabling encryption and determining your exposure to peers. Queuing settings and seeding limits can also be set, with conditional options for how to treat a completed torrent.

The RSS tab is for hardcore downloaders, and is provided for bulk torrent file management.

As noted, the Web UI tab lets you manage authentication of the web interface. You can also tweak language, enable HTTPS (with a certificate), and control some security options. If you plan to access from beyond your home network, the web UI supports dynamic domain names from No IP and DynDNS.

Finally, the Advanced tab covers everything from announcing new torrents to trackers to managing disk caching. You won't need to access most of this but the web UI can launch documentation if you do.

What do all the numbers mean?

If you've used BitTorrent clients in the past, you've probably spotted that the user interface is typically a

bit complicated. You might be interested purely in whether the ISO or whatever you're downloading is going to complete soon, but there is actually quite a bit of information available.

You'll usually find that the name of the torrent is followed by its size. This is useful in various ways: it lets you know how much space it will take up on disk, and it gives you a general idea of how long the file will take to download. It can also help you to identify whether the file is genuine; for example, an Ubuntu ISO isn't going to be 20MB. Fake files distributed by scammers remain a problem on the BitTorrent network, so beware.

You'll see a percentage in the Done column, which represents the completion of the download. After the Status column (which typically displays 'Seeding' or 'Stalled') you should see a series of numerical columns.

Seeds lists how many people – like you – are sharing the file. Peers are the people/torrent clients attempting to download and share what has been downloaded so far. Generally, more peers means a faster download.

Down Speed lists how fast a torrent is being downloaded; Up Speed is the rate of upload, interesting to anyone hosting an ISO and uploading it. ETA doesn't apply to uploads, but proves a useful guide to when you can expect a download to complete.

Finally, the Ratio demonstrates how much you have contributed to sharing the file. The higher this figure, the greater your reputation.

Automatically start qBittorrent

One last thing you might want to do with your torrentbox is to ensure it automatically launches when your Raspberry Pi boots up. You can do this by editing the **crontab** file:

\$ crontab -e

Select an editor (Nano is the most usable option) and use your keyboard's arrow keys scroll to the end of the file. Here, add:

@reboot/usr/bin/qbittorrent-nox

Press Ctrl+X to exit, and Y to save. Now when the Pi reboots, *qBittorrent* automatically loads, enabling torrenting to continue with no input from yourself.

With your Pi-hosted torrentbox up and running, you can download and share files as needed. Unsavoury uses of the technology aside, hosting open source ISOs is a nice way to help the FOSS community and use the low power of your Pi to pay something back.

QUICK TIP

The more you share, the more storage you need. Don't rely on an SD card if the ISOs are too large, or you have multiple files to share; hook up some USB-based storage instead.

>> STAY SAFE

The BitTorrent network has a bad reputation for piracy, sleaze and malware. While this guide provides the tools to create a torrent of an open source ISO and share it, you might be tempted to look into things in more depth.

You will inevitably find yourself on the bad websites. What do you do? Well, short of avoiding them entirely, you should be aware of the risks to your browser. Malicious adverts (malvertising) are a problem, along with attacks that use vulnerabilities in browser plugins.

Ensuring your browser is up to date before visiting these sites is smart. But as plenty of legal torrents are available, there is no reason to leave the path of righteousness.

Instead, check out Public Domain Torrents (https://bit.

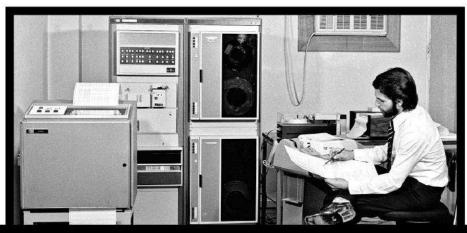
ly/LXF311-PDtorrents) or the Internet Archive (https://bit. ly/LXF311-archive), which hosts various legal torrents.

BitTorrent activity may be blocked by your ISP. This is an unsurprising consequence of people misusing technology to break copyright. As such, you may find that you cannot seed your chosen ISO file(s).

The smartest solution to overcoming this roadblock is

to use a VPN. Additionally, you need a paid VPN service that supports BitTorrent and peer-to-peer networking; free VPNs usually don't do this, and those that do don't provide adequate bandwidth.

To use a VPN on a
Raspberry Pi, you need one
that supports OpenVPN.
Then use the .ovpn profiles
that your VPN provider issues
for a private connection.



HP was one of the key minicomputer manufacturers of the '70s to offer Basic, specifically on its 2100 Series machines

The rise and fall (and rise again) of Basic

We plot the symbolic milestones of the most basic of allpurpose code with the instruction of **Mike Bedford**.

ere at Linux Format Towers, we're always exploring programming languages, some of which date back to the late '50s but still play important roles today. Those venerable languages are Fortran and COBOL. Here we're engaging on a similar exercise, as we turn our attention to Basic, but the story is different. The timeline of Fortran and COBOL can be summarised as: developed back in the mists of time; enjoyed almost immediate success; accepted as a standard; several new standardised versions are introduced over the decades; currently within the top 25 spots in the TIOBE index; widely used in research and/or business today. In the case of Basic, while the first three and last two of those statements remain true, the journey is substantially different, as we're about to see.

Although this article can be considered a history lesson – albeit hopefully more enthralling than learning the dates on which British monarchs ascended to the throne – there's a hands-on element, too. Of course, it's quite likely that you've already tried your hand at Basic programming, but there are loads of variants, so hopefully we'll introduce you to something new and unexpected. And

in one of its newer guises, you might even decide to use it for coding a real-world app.

Despite its longevity, there's no way that Basic is going to challenge the likes of Python, Java, C and C++. Indeed, we'd be surprised if a few hackles aren't raised at our suggestion that Basic could have any place in the world today. After all, it has come in for its fair share of criticism over the years, mainly because of the potential, in its earlier versions, it was argued, for it to promote poor programming practices. Love it or hate it, though, it's surely good to understand this important part of our programming heritage.

The early days

Basic was launched in 1964, making it only the fifth mainstream high-level language, after Fortran, ALGOL, LISP and COBOL. And like its predecessors, it carved out its own niche. In particular, it was designed as a language for teaching beginners to program, indeed the acronym – Beginners' All-purpose Symbolic Instruction Code – spells that out quite clearly. Created by John Kemeny and Thomas E Kurtz at Dartmouth College in New Hampshire, not only was the syntax simple, but it was also implemented on a time-

50 **LXF311** February 2024

sharing system, enabling multiple users to work simultaneously. The system allowed students to enter, edit, list and run their code from remote teletype terminals. This was a far cry from submitting code as a stack of punch cards, and receiving printed output several hours later, a model that was almost universal. This primitive form of IDE would be widely adopted elsewhere as Basic migrated to other computers. However, the Dartmouth system actually compiled the code in its entirety when the user ran it, while many other Basics were implemented as interpreters, compiling each instruction as and when it was run.

Dartmouth Basic could be described as Fortran-like – albeit simplified – and was quite different from many of the more recent versions of the language. Instructions were one per line, and those lines had to start with numbers. The numbers made it easier to edit the code – a statement could be replaced by entering a new statement with the same line number, or deleted just by typing its line number – and the numbers were also used to define the target of instructions such as **GOTO**. We can see all of this in the following sample code, which also allows us to investigate some of the instructions that were available.

10 REM CELSIUS TO FARENHEIT CONVERSION 20 DATA -300, -10, 0, 5, 10, 20, 100, 200, 500, 1000 30 FOR N = 1 TO 10

40 READ C

50 IF C >= -273 THEN 80

60 PRINT C, " CELSIUS IS BELOW ABSOLUTE ZERO" 70 GOTO 100

80 LET F = C * 9 / 5 + 32

90 PRINT C, "CELSIUS EQUALS ", F, " FARENHEIT" 100 NEXT N

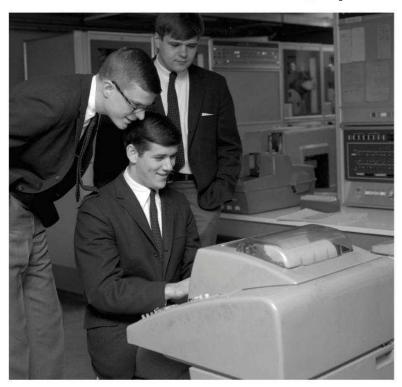
110 END

A few things in this program probably appear rather strange, first and foremost being that it's all in capitals. That was common practice in the early days, though, because of the limitations of terminals, card readers and printers. Next, unlike the vast majority of other languages, the assignment statement isn't of the form X = 10, X := 10 or similar. Instead it has a keyword associated with it, namely LET, so the previous example would be LET X = 10. And the concept of everything having an associated keyword even extends to comments. Rather than being prefixed by one or two characters such as # or //, in Dartmouth Basic

comments start with **REM**, which is short for remark. A few words about the **DATA** and **READ** statements are also appropriate. There was no input statement to ask the user to enter a value or to read data from file. Instead, data was provided using one or more **DATA** statements in the code, each of which was followed by a list of values, and these were read using the **READ** statement. Also of note is the **IF** statement, which is unusual, by today's standards, in only defining an implied **GOTO** if the condition was true.

Moving beyond the sample program, Dartmouth Basic had 15 instructions, specifically **DATA**, **DEF** (userdefined function), **DIM** (dimension an array), **END**, **FOR**, **GOTO**, **GOSUB** (execute subroutine), **IF**, **LET**, **NEXT**, **PRINT**, **READ**, **REM**, **RETURN** (return from subroutine) and **STOP**. There were also 10 built-in functions, namely **ABS**, **ATN**, **COS**, **EXP**, **INT**, **LOG**, **RND**, **SIN**, **SQR** and **TAN**. Note that variable names were rarely meaningful as they could only be either a single letter or a single letter followed by a single digit, and array names were limited to just a single letter. If you want to

Running on a small General Electric GE-225 mainframe, Dartmouth's multi-user Basic system made programming more accessible to beginners.



WHAT? NO COMPUTER?

Perhaps one of the strangest dialects of Basic is weird in no small part because it doesn't run on a computer, at least not one in the normal sense of the word. Instead, it was designed to run on a scientific calculator. We're talking here of Casio Basic, which was developed to allow user-defined functions to be used, although it's also been used to implement

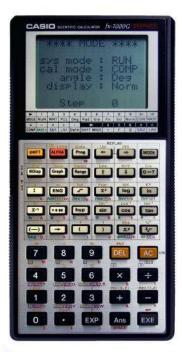
games. What's more, if you thought this refers to the dim and distant past, you can still buy Casio calculators today, complete with Casio Basic.

It's fair to say that Casio
Basic is one of the more
distant relatives of Basic. It's
a very basic Basic, although it
does have some post-1964
features, such as a while
loop and an else option to
the if statement. In terms of

variables, it's very limited, offering just 26 names, the letters A to Z. It also departs from Dartmouth Basic's use of **LET** for assignments, and not just by omitting that keyword. Instead, it uses the → symbol, for example 10 → A. Obtaining user input is just as terse, for example "ENTER A NUMBER"? → A, as is printing results to screen, for example B Sometimes,

though, this abbreviation results in more readable code, as √A is surely an improvement over SQR(A). And the reason for making the code as abbreviated as possible? It probably has more than a little to do with the fact that some Casio calculators didn't have QWERTY keyboards, so multiple keystrokes were needed to enter letters.





Basic even appeared on calculators, like this 1985 Casio fx-7000, although it has sometimes been described only as "Basic-like".

The Commodore 64 was the best-selling home computer, by a significant margin. So it seems likely that its Basic was the most-used dialect during the '70s and '80s home computer boom.

know more, take a look at the very first user manual, which is available at www.dartmouth.edu/basicfifty/basicmanual_1964.pdf. The biggest criticism of this and other early Basics was that flow control was limited to GOTO, GOSUB, IF and FOR ... NEXT. In omitting almost all forms of structured code, it was argued, beginners were taught poor programming practices, which were hard to unlearn. It has to be said, though, that many a professional programmer gained their first experience of coding via Basic with no apparent ill effects.

We struggled to find a FOSS implementation of the original Dartmouth Basic. However, to try out some true 1964 Basic code, you might think that all you need to do is find an implementation of Basic that preserves the use of line numbers, then just discipline yourself not to use

the more recent extensions that might be provided in your Basic version of choice. But that's easier said than done, because many modern Basic implementations don't have all the instructions that were available in Dartmouth Basic. Ideally, we'd have liked to suggest an online resource, since Dartmouth Basic is surely only going to be a curiosity and you might not want to install anything locally. We didn't really find anything suitable, though, so how about installing Bywater Basic, which is called bwbasic in repositories? When you enter the sample code, bear in mind what we said earlier about replacing and deleting lines, and you should also know that you can see your complete code by typing LIST. And if you move on to writing your own code, it was normal practice to use line numbers separated by 10, so you could insert new statements later. We do have to report what we believe be an error in Bywater Basic - our sample code reads the data in the wrong order, with -300 being last. If this offends you, try the online www.quitebasic.com, but you have to depart from Dartmouth Basic by swapping all the commas in the PRINT statements for semicolons, and bear in mind that the IDE is much more modern than that provided by Dartmouth Basic's archaic command-line system.

Versions, versions and more versions

In looking at Fortran and COBOL over the decades, standardisation was a common theme. Both languages became official standards in their early days, and new versions of those standards, each introducing new



features, continued almost to the current day. At one time it looked like Basic might have followed a similar path, but things didn't turn out that way.

The original Dartmouth Basic was updated several times by its initial developers, as might have been expected. And once it reached a reasonable degree of maturity, attention did indeed turn to standardisation. Unlike Fortran and COBOL, though, it didn't become one standard but two. First there was Minimal Basic, based on an early iteration of Dartmouth Basic, which was adopted as an ANSI standard in 1977. And second, there was Full Basic - which added string manipulation functions, support for structured programming concepts, input/output for file handling and more which became an ANSI standard in 1987. But unlike the case with many other languages, these were not followed by Minimal Basic 2, Full Basic 1990 or the like. In addition, it seems hardly any standards-compliant products were ever released but, instead, non-standard versions of Basic proliferated. Indeed, many were released prior to the ratification of Minimal Basic, and perhaps most came on the scene before Full Basic was approved, illustrating the irrelevance of such standards.

The first generation of vendor-specific nonstandard Basics appeared in the late '60s and early '70s. Produced for minicomputers - which had more limited hardware resources than mainframes - these Basic variants catered to those machines' limitations. There were also benefits from abandoning the standard - manufacturers could pick their own extensions to favour their user base. One of the first such products was HP Time Sharing Basic, which aimed to replicate the time-sharing approach - but not the exact syntax - of the original Dartmouth system. It ran on HP's 2100-series minicomputer, in the guise of the 2000-series, which was a main minicomputer bundled with a smaller 2100 machine that acted as a terminal server for the multi-user Basic system. Other versions of Basic for minicomputers included ones for the DEC PDP-8, PDP-11 and VAX-11 series, the Wang 2200 series, and the Data General Nova. These systems were used in academia, preserving Basic's heritage as an educational language, but things were changing. Fuelled by the lower cost of computer ownership that minicomputers heralded, and the easeof-use benefits that Basic provided, the language started to be targeted to industry and business, too. In fact, Data General's Business Basic was even considered to have been inspired by that archetypal business language of the day, COBOL.

Dawn of the PC

And so we come to the era of the microcomputer, or PC, to use the word in its broadest sense. Starting in the mid-'70s and running through to the mid-'80s as low-cost machines kick-started computing for enthusiasts, Basic took off big time. Probably the first version to appear on a machine aimed at hobbyists was Altair Basic, developed by Microsoft and appearing in 1975. It would go on to become Microsoft Basic and influence many other Basic dialects, but it was written for the MITS Altair 8800 which many consider the first enthusiasts' computer, albeit a hugely expensive one.

As we come to the late-'70s and early '80s, the era of home computers starting at less than £100, Basic

52 **LXF311** February 2024 www.linuxformat.com

versions mushroomed. It would take for ever to discuss them all, but Basic dialects were launched by the following companies, to name but a few: Acorn (BBC Micro), Amiga, Atari, Commodore, Dragon, Sinclair and Tandy (TRS-80), and some of these spawned multiple variants for different models. Why so many, you might wonder? The main reason is that home computers didn't follow standards, having unique hardware, most notably their displays. So, assuming that manufacturers wouldn't have been happy with a Basic that worked only at a textual level – and that wouldn't have met the needs of '80s home computer users – they had to customise the language to support their graphics.

Then we come to the family of more application-specific versions of Basic that were intended for game development. AMOS and Blitz Basic did this job for Amiga, while STOTS catered to Atari users, and this class of software has been described as predecessors of today's game engines. But supporting different platforms and simplifying game creation was only part of the story of home computer Basics. Support for structured programming, elimination of line numbers, support for lower-case letters and object orientation all made their appearances. If you want to know more, we've featured several of these dialects in LXF over the years, including BBC Basic only last month.

Decline and rebirth

Basic on home computers introduced vast numbers of people to the wonders of programming. But this craze didn't last long. Programming on early '80s machines was largely a necessity, because of the relative scarcity of off-the-shelf software. Needless to say, it didn't take long for software vendors to cater for this need and, in so doing, diminish interest in coding.

As Basic declined on enthusiast-orientated computers, though, it saw a resurgence on Microsoft-powered PCs, and this continues to the present day. What's more, it didn't remain stuck in the world of Windows. For although you won't find anything by the name of Basic occupying a prominent position in the TIOBE index, you will find Visual Basic in the 21st spot. Introduced in 1992, this object-orientated language improved productivity in creating Windows applications by allowing GUI elements to be defined graphically, rather than by manually entering their



defining parameters. Needless to say, this hasn't been ported to Linux; in fact, it couldn't be, since it's so closely tied up with Windows. However, a language with exactly the same aims and similar syntax, but running under Linux and building applications using the Ot or GTK toolkits, exists in the form of Gambas. The Gambas wiki lists 85 applications that it describes as "medium or big applications written in Gambas". However, to really see the penetration of Visual Basic in the world of Linux, we need to look elsewhere. If you have an open office suite installed on your PC - and with most distros it's pre-installed - you already have a means of writing code in a language that was heavily influenced by Visual Basic. Both Apache OpenOffice and LibreOffice support a Visual Basic-like scripting language for macro development, indeed it's the default language in the latter.

So, it might be the language that refused to die, but are you really coding in Basic when you write a macro in LibreOffice? Certainly, a Basic programmer timewarped from 1964 wouldn't recognise LibreOffice Basic as such. If we wanted to be mischievous, we might just be tempted to describe today's dialects of Basic as like Grandma's broom. If you're not familiar with that concept, it refers to an old lady who reportedly said, "I've had this broom for the last 70 years, and in all that time it's only had six new heads and three new shafts."

Huge differences in hardware among home computers – especially the graphics – were responsible for a massive proliferation of Basic variants.

>> BASIC GETS SERIOUS

We're used to the idea of Basic of old being either a beginner's language or one for enthusiasts, but it's interesting to consider whether it was ever used for serious product development in its pre-Visual Basic forms. Anecdotal references of such aren't in short supply but, more definitively, we can report on at least one major software product that was written in classic Basic. And

given that the software company responsible was estimated as the world's seventh largest in 1984, we find it hard to believe that it was an outlier in these early days of the IBM PC.

The company in question was called Peachtree Software, and its product range is now owned by accountancy software company Sage. Back in 1981, though, Peachtree brought

the world's first entry-level accounting software to the market, initially for personal computers running the CP/M operating system. Founder Ben Dyer, talking to the Center for the History of Information Processing of the Charles Babbage Institute in 2004, revealed some interesting facts. First, and most importantly, he confirmed that their first products were indeed written

in Microsoft Basic. And it gets more interesting when we learn just how important this software was to the embryonic business PC market in the early '80s. According to Dyer, IBM had identified Peachtree's accounting software as a principal product for the IBM PC, approached Peachtree, and ended up providing funding for its development to run under MS DOS on PC.

TUTORIALS



Total shell recall

Too busy running around shouting, "Get your ass to Mars," **Shashank Sharma** thinks a strong shell history can be a real life saver.

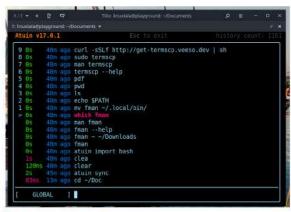


Shashank Sharma is a trial lawyer in Delhi, India. He's been writing about open source software for 20 years and lawyering for 10.

hell history is one of the most useful and powerful tools at your disposal if you spend a lot of time working with the CLI. While you can easily tweak the settings on your preferred shell to store 500, 2,000, or even more commands, *Atuin* offers something far superior. It replaces the traditional shell history with a SQLite database, but more importantly, it gives you the ability to sync your CLI history across different machines.

Once installed, Atuin rebinds the Ctrl+r and the up arrow key, used to reverse look up previously executed commands and access the history respectively, with a full-screen history search interface. You can also get stats on the most executed commands, or look up commands based on exit codes and more. Best of all, Atuin doesn't replace or overwrite your existing history file, so you can revert to it at any time should you find Atuin not to your liking.

Released under the MIT licence, Atuin is a breeze to install if you already have Rust and Cargo installed and



For our money, we prefer to use Atuin without the Ble.sh plugin, but try it yourself and see if you like its autocompletion.

configured on your distro. Just execute the cargo install atuin command. The project also offers an installation script, which uses *Cargo* if available or *Homebrew*, depending on how your distro is configured. You can find detailed instructions on the project's website (https://atuin.sh) and even the GitHub page (https://github.com/atuinsh/atuin).

>> ENABLE SYNC

If you would prefer to sync your history, you can either use *Atuin's* server to store your history, which is end-to-end encrypted, or host your own server. To use *Atuin* to host your history, run the atuin register -u <username> -e <email> command. You're prompted to enter a password and then *Atuin* generates a key. This key is used by all other machines to log in to the *Atuin* server to sync the history. The encrypted key is stored in ~/.local/share/atuin/key file but you can run atuin key to view the unencrypted key at any time.

Once you've registered an account and obtained the key, you can log in to the *Atuin* account on all other devices with the atuin login -u <username> command. You're prompted for the encrypted key file and the set password. Once you provide both, *Atuin* starts syncing the history.

By default, *Atuin* syncs the history hourly; however, you can force a sync at any time with the atuin sync command. You can define a custom sync setting by editing **sync_frequency** in the **~/.config/atuin/config.toml** file.

You can alternatively host your own server if you'd rather not use *Atuin's* public server. This requires PostgreSQL 14 or higher, and a very minimal configuration. Refer to the project's documentation for config instructions (https://atuin.sh/docs/self-hosting).

Installation and config

If you don't already have Rust and *Cargo* installed, you can use your distro's package manager to install these. The <u>sudo dnf install rust cargo</u> command installs these on RPM-based distributions, such as Fedora. You can similarly run the <u>sudo apt install cargo</u> command, which also installs the **Rustc** package, if you're running Ubuntu or Debian, or one of their derivatives.

Apart from the *Atuin* package, you also have to install a shell plugin, either *Bash Line Editor*, aka *Ble.sh*, (https://github.com/akinomyoga/ble.sh) or *Bash Pre-Exec* (https://github.com/rcaloras/bash-preexec) if you work with the *Bash* shell. The *Atuin* project, recommends *Ble.sh* for optimum performance.

However, this isn't a strict requirement, and you can use *Atuin* even without *Ble.sh*. If installed, *Ble.sh* replaces the default GNU Readline in the *Bash* shell and provides many functionalities such as syntax highlighting, enhanced autocompletion, and so on.

While Atuin and Ble.sh offer installation instructions, you can copy the following steps to install and enable

54 LXF311 February 2024 www.linuxformat.com

QUICK TIP

key should be

stored safely,

preferably in

a password

manager or

other similar

shared with

anyone. You

might even

consider using Enc, a project

we discussed

in LXF310 for

this purpose.

way to recover

the key using

Atuin. Your only

recourse would

be to register

a new account

and resync the devices using

the new key.

There's no

tool, and never

The Atuin

Atuin if you work with Bash and already have Rust and Cargo installed:

\$ cargo install atuin

\$ git clone --recursive --depth 1 --shallow-submodules https://github.com/akinomyoga/ble.sh.git

\$ make -C ble.sh install PREFIX=~/.local

\$ echo 'source ~/.local/share/blesh/ble.sh' >> ~/.bashrc

\$ echo 'eval "\$(atuin init bash)"' >> ~/.bashrc

\$ source ~/.bashrc

With these commands, we've first installed *Atuin*, and then downloaded and installed *Ble.sh*. We next added entries to the ~/.bashrc file so that *Ble.sh* and *Atuin* are loaded every time you launch *Bash*.

You can now opt to populate *Atuin* with your existing shell history. This is done with atuin import shell). Run atuin import bash if you use the *Bash* shell, or atuin import zsh if you use *Zsh*. You can also run atuin import auto, and *Atuin* determines your current shell and imports the relevant history file.

Atuin stores relevant files such as history in the ~/.local/share/atuin directory, while the configuration is stored in the ~/.config/atuin/config.toml file. You can open this in your favourite text editor and make changes as necessary. We recommend you spend some time with the official documentation (https://atuin.sh/docs) before you start tweaking the settings.

The setup we've discussed here is all you need to do if you prefer to use *Atuin* in offline mode. That is, your shell history is stored but it isn't synced to be shared across different devices.

Refer to the Enable Sync box (opposite) if you wish to share your shell history across multiple devices.

Meaningful search

Atuin defaults to a fuzzy search, but you can tweak this by editing the **search_mode** setting in the **~/.config/atuin/config.toml** file. Your options are fuzzy, prefix, fulltext or skim.

The Atuin search interface, accessed by pressing Ctrl+r or the up arrow, lists the total number of commands in history at the top-right. By default, Atuin performs a Global search, but you can press Ctrl+r again to switch to Host, Session or Directory search. That is, the search can be restricted to the current device instead of all devices, only the current session or only the current directory. You can switch between these search modes by pressing Ctrl+r.

You can also search for commands with an exit code, to identify commands that executed successfully. The atuin search --exit 0 <search-term> command lists commands that were successful, whereas atuin search --exclude-exit 0 <search-term> lists commands that failed.

You can also use switches such as --before and --after to define time limits when performing a search. The --limit switch can be used to restrict the number of results produced by Atuin. The atuin search --limit 5 'dnf install' command, for instance, lists the last five packages installed using dnf.

The --reverse switch can similarly be used to find the oldest matching result for your search query, and you can even combine the different switches to refine your search. So, to find the first package you installed using dnf, you can run the atuin search --limit 1 --reverse "dnf install" command.

You'll notice that the *Atuin* search UI provides a number from 0–9 for the commands listed in the interface, The numbers move up and down as you scroll through the history. You can select a command at any given number by pressing Alt+<num>. (Refer to the screenshot, *left*, for reference.) During our tests, this feature worked flawlessly on our Fedora machine but not on our Ubuntu installation.

For a list of your most frequently used commands, you can run the atuin stats command. Atuin supports a number of subcommands, such as search, stats, key, sync and so on. Run the atuin -h command for a complete list of subcommands.

Many of the subcommands have their own switches, so run the atuin <command> -h command if you need help with those. For instance, the atuin search -h command can be a good introduction to how to get the best results with Atuin.

For a list of different devices sharing *Atuin* history, you can run the **atuin status** command:

\$ atuin status

Atuin v17.0.1 - Build rev

[Local]

Sync frequency: 10m

Last sync: 2023-12-07 11:19:40.166157135 +00:00:00

History count: 1177

Deleted history count: 0

[Remote]

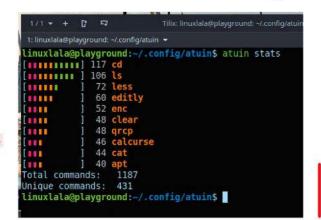
Address: https://api.atuin.sh

Username: linuxlala

History count: 1176

Because Atuin doesn't alter your existing shell history, you can run the atuin account delete command to delete your account on the Atuin server without affecting the local history.

The project is actively being developed, with many additions and enhancements recommended by users on the Issues page (https://github.com/atuinsh/atuin/issues), such as the option of changing the password for the Atuin account. Should you run into a problem, visit the Issues page to see whether it has already been discussed and resolved by the community.



We're delighted to see that our love for nInvaders and nSnake hasn't been outed.

>> ENHANCE YOUR TERMINAL-FU Subscribe now at http://bit.ly/LinuxFormat

MOTION Credit: https://motion-project.github.i

Build your own home CCTV system

Nick Peers reveals how to bypass expensive (and insecure) cloud security systems to create your own using a variety of cameras.



Nick Peers
has lost hours
trying to get
cheap knock-off
cameras to work
with MotionEye.
Ironically his
greatest success
came with his
oldest one.

orried about your home security? Looking for a system that lets you set up multiple cameras around your property, but which won't break the bank or require you to invest in an expensive monthly subscription while entrusting your video footage to some unknown third-party cloud hosting service? The obvious solution is to invest in a dedicated NVR - network video recorder but why tie yourself into a single expensive proprietary ecosystem when you can cannibalise cameras from a variety of sources and use a self-hosted solution that's completely free and private?

The good news is that your Linux-powered PC or server stands ready and willing to shoulder the NVR burden. There are several open source platforms out there, but here we're focusing on *Motion*, a long-established and completely free option that you can install on a dedicated machine like the Raspberry Pi or add to your server.

There are several versions of *Motion* out there, but we're featuring *MotionEye*, which offers a web-based front-end you can manage from any web browser, and which can be installed in a minutes through *Docker*. Better still, *MotionEye* does everything you need of it not only can you attach multiple cameras for viewing and recording your home in real time, but you can configure it to jump into action whenever motion is detected, recording potentially incriminating footage while alerting you via email.

In this tutorial, we'll show you how to get *MotionEye* up and running, link cameras to it from a variety of sources, set up motion detection and enable you to access the service from outside your home.

Spot a camera

Before setting up *MotionEye*, ask yourself what cameras you'll need. The good news is that you don't need to lay out hundreds of pounds on expensive cameras to populate your surveillance system. *Motion*



We recommend setting up your Docker script in a text editor to make it easy to tweak and redeploy when required.

works with a wide range of cameras and video devices, including USB webcams, video capture cards and – crucially – a wide range of wireless IP security cameras. You can even breathe new life into old phones and tablets by turning them into IP cameras.

Sadly, you can't just buy any old IP camera and expect it to work out of the box. The most crucial thing is that the camera is RTSP or ONVIF-compliant. But don't take any claims at face value, particularly with cheap Chinese models that flood eBay listings – some cameras won't work full stop, while others require specific firmware to unlock ONVIF support, which can be next to impossible to find (and even if it's currently in place, may be wiped out by a future update).

Start your search at www.ispyconnect.com/
cameras for a list of cameras that should potentially
work with MotionEye. These should reveal the allimportant RTSP information you need to feed to
MotionEye so it can connect to the camera in question.
The hardest trick is linking these to the myriad of
cameras offered on eBay.

Take the V380 Pro series, for example. This label covers a wide range of cameras that bear very little resemblance to each other. Some models – including one we sourced a few years ago – support the RTSP/ONVIF protocol, but it's disabled by default. Visiting https://gist.github.com/SolveSoul and clicking the

QUICK TIP

If you feel 'admin' is too username for your main account, after setting your password, browse to the MotionEye configuration folder and open motion.conf in a text editor. **Uncomment the** #@admin_ admin line and change admin to your choice of username. Save the file the change is immediately applied.

56 **LXF311** February 2024

Ceshi.ini link reveals instructions on how to use a simple script to unlock ONVIF capabilities – it worked perfectly in our case, but check the comments to see how many newer models no longer work with this simple hack.

If you already have some IP cameras, you can quickly determine if they'll work with *MotionEye* using another *Docker* container: *Cameradar*. All you need do is apply a single command (substitute the **x** in **192.168.x.0** with that used by your local network):

\$ docker run -t ullaakut/cameradar -t 192.168.x.0/24

Cameradar scans your entire network for cameras capable of broadcasting over RTSP, then lists any ones it finds with all the details you need (URL, plus whether you need a username and password) to access it through MotionEye.

Motion to install

You're now ready to install *MotionEye* on your PC. While a standalone build exists (see https://github.com/motioneye-project/motioneye/ for details), it's not compatible with Ubuntu 22.04 or later. We're therefore going to deploy it using *Docker*, which has the added benefit of making it easy to set up for remote access through a remote proxy like *Nginx Proxy Manager* (see the box on page 59).

We suggest copying and pasting the full script (see screenshot, left) from https://github.com/motioneye-project/motioneye/wiki/Install-In-Docker into a text editor, then amending it as instructed. This may include the ports that MotionEye needs – 8765 for accessing the web interface, and 8081 onwards if you wish to view each camera's stream independently of MotionEye. As always with Docker containers, you can map these to different ports if they clash, like so:

-p 8766:8765 -p 9081:8081 -p 9082:8082 -p 9083:8083 \

You also need to insert an extra line beneath the -p line to join *MotionEye* to the same shared bridge network as your proxy server's *Docker* instance if you're planning remote access – change **shared-network-name** accordingly:

--net=shared-network-name \

If you plan to connect any webcams plugged directly into your PC/server via USB, then add one or more lines for each webcam like so:

- --device=/dev/video0 \
- --device=/dev/video1 \

Finally, you'll need to set up a configuration folder for your *MotionEye* preferences, which you'll point to



Adding an RTSP or ONVIF camera requires you to know what URL (and RTSP port) the camera communicates on.

>> TURN YOUR PHONE INTO A CAMERA



If you've held on to your old phones and tablets, *MotionEye* may well be the tool you need for giving them a productive second life. All you need is a suitable app to convert the mobile into an IP camera – it's free on Android, and costs \$2.99 on iPhone.

Pavel Khlebovich's *IP Webcam* (https://play.google.com/store/apps/details?id=com.pas.webcam) runs on any device running Android 4.1 or later, while *ipCam* (https://skjm.com/ipcam) runs on any iPad, iPhone 5S or iPhone Touch 6 or later (iOS 12.0 required).

Both are relatively simple to configure. Simply enable ONVIF support in *IP Webcam* and then start the server – the URL you need when connecting your phone as a Network Camera in *Motion* is rtsp://192.168.x.y:8080/h264_ulaw.sdp – take the time to add username and password protection to the feed in the app's settings.

With *ipCam*, the URL you need when setting up the camera as a Network Camera is **http://192.168.x.y/video.mjpg** – again, use the app's options to configure password protection for extra security.

Once configured, all you need to do is find a suitable location for your phone, one that's close to a power socket to ensure your mobile doesn't die at a crucial moment, plus a stand to place it on. You'll find numerous options available on Amazon and eBay, but if your budget's tight, you could fashion a simple stand out of everyday objects, such as a paperclip or expired credit card, as shown on WikiHow (www.wikihow.com/Make-a-Phone-Stand).

/etc/motionEye in the *Docker* script, as well as a folder to store any video recordings, which will point to /var/lib/motionEve.

Once the script is configured correctly, select it all and copy it to the clipboard. Now open the terminal and press Shift+Insert to paste the *Docker* script into the terminal. Hit Enter, and all being well, *MotionEye* should start and be available within seconds.

Rapid eye setup

Going forward, all interaction with *MotionEye* is done through any web browser on your network. Type http://192.168.x.y:8765 in the address bar, substituting 192.168.x.y with your PC's IP address and changing 8765 if necessary. You're prompted to log into the *MotionEye* web-based UI – type admin into the username field, leave the password blank and hit Enter.

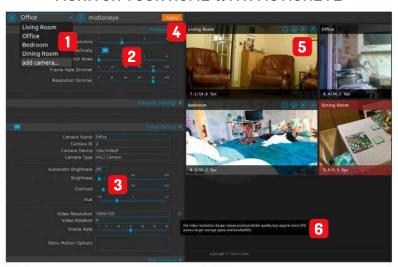
You'll see a prompt to add your first camera, but resist temptation and instead focus on securing the web interface. Start by changing the password: click the hamburger button in the top left-hand corner to reveal the main *MotionEye* menu panel. You'll see fields for setting both an admin user and password, and a

QUICK TIP

Before opening vour server to internet access. note that anyone can view your camera feeds simply by visiting your domain or subdomain. To passwordprotect the feeds so only you (or trusted friends and family) can see them, take the time to set up a Surveillance user under General Settings.



MONITOR YOUR HOME WITH MOTIONEYE



Manage cameras Click the drop-down menu to reveal a list of cameras enabling you to switch, add or remove (click the bin icon) them.

Arrange camera layout Use these controls to change how the various camera feeds appear - use the dimmers to save bandwidth.

Camera controls You get full control over a USB webcam's brightness, contrast and hue; IP cameras limit you to auto-brightness only. Apply changes

Don't forget to regularly click the Apply button to perform any changes or tweaks the entire view updates when you do.

Access recordings

These buttons enable you to view that feed full-screen, view recorded images and videos, plus access its settings.

Context-sensitive help Not sure what a setting does? Roll your mouse over it and a? icon appears - hover the mouse over it for an explanation.

view-only surveillance user and password - see the Quick Tip (page 57) for why you might want to set this latter user up.

Start by setting a password for the main admin user - once entered, hit Enter, then click the orange Apply button that appears at the top. You're immediately logged out - log back in using the admin username and the password you just set. If you want to set a more secure username, check out the Quick Tip (page 56).

If you're

QUICK TIP

desperate to try to get a cheap Chinese IP camera working, vou'll find two forums in particular worth investigating: Use-IP (www. use-ip.co.uk/ forum) and **IPCamTalk** (https:// ipcamtalk. com/forums). Also try to get as much information about your IP camera as possible use its app to determine make, model and firmware

Add your cameras

Now click You Have Not Configured Any Camera Yet, Click Here To Add One to open the Add Camera wizard. By default, Local V4L2 Camera is selected, and if you have any USB-attached cameras, they appear here. Click OK to add the camera if so. If not, click Local V4L2 Camera to open a drop-down menu offering a choice of inputs. Local

MMAL Camera refers to the Raspberry Pi camera module, while Remote MotionEye Camera would enable you to connect to cameras already attached to another MotionEye instance on your network.

For most people, therefore, it's a strange choice between Network Camera and Simple MJPEG Camera. RTSP/ONVIF cameras require Network Camera. Enter the required URL, which if you're lucky is as simple as rtsp://192.168.x.y (your camera's IP address as revealed by ONVIFViewer), then press Tab. If the connection is successful, the Camera drop-down should change to RTSP/TCP Camera (with an RTSP/UDP Camera option if

applicable). Enter the username and password required to access your camera and hit OK.

If successful, after a short pause you should see your camera's feed appear in the main window, along with a panel offering you an array of configuration options. Let's explore these before leaving you to set up any subsequent cameras you own.

Eyeballing your config

Use the Preferences section to configure the layout of the main MotionEye screen - by default it's configured to show three feeds per row, so set Layout Columns to 2 to increase the size of each feed on screen.

Next, scroll down below General Settings, where you'll find a series of categories related to the configuration of this, your first video feed. Start with the Video Device section. Begin by renaming the camera to something more descriptive - typically its location for easy identification.

Below this you'll see the camera's streaming settings in terms of resolution and frame rate. MotionEye deliberately sets these conservatively at just 640x480 and 2fps - jerky, but good for lowbandwidth connections. Click the Video Resolution drop-down menu and nudge the Frame Rate slider to the right to increase these values, but avoid the temptation to push them to their maximum (say 1,920x1,080 @ 30fps). You need to balance the number of cameras you're adding with your PC's capabilities and your network connection. You'll want to experiment with different settings, but for now try a more median figure - say 1,280x720 @ 15fps. To see the effects of your changes, click Apply and wait for the screen to refresh.

Further configuration

Beneath Video Device you'll find eight more sections to explore. File Storage enables you to alter the default storage path and see how much space is on your hard drive, but most people should leave these settings untouched. Beneath this is Text Overlay, which is where you can alter the text that appears over the video - by default, it's the camera name and a timestamp, but the most important thing is to make it more visible using the Text Scale slider (3 or 4 is a comfortable minimum).

Beneath this is a Video Streaming section, enabled by default. This allows you to set up web-based live



Recordings are stored in the universal MP4/H.264 format - these can be downloaded, then deleted to free up space.

ersion.

CONFIGURE MOTION DETECTION



To ensure that any detection
To ensure that any detected
movement is recorded, expand the Movies
section and verify Motion Triggered is
selected under Recording Mode. Also
verify that the most compatible file type
(MP4 format with the H.264/OMX codec)
is selected under Movie Format. Tweak
other settings to fit your needs.



Set motion sensitivity
Now expand Motion Detection. Roll
your mouse over the? next to each item
for a detailed description, but focus first
on Frame Change Threshold, which
determines how sensitive the camera
is to motion detection, with 0 the most
sensitive. Experiment with different levels
between 2 and 20 to find a happy medium.



Set an image mask
To apply motion detection to only
part of the video, flick the Mask to On.
The Smart Mask Type learns over time,
or you can set this to Editable and click
Edit Mask to manually select the areas
you would like to exclude from motion
detection using the grid. When done, click
Save Mask followed by Apply.

streams and snapshots that you can access directly using specific URLs, then set various quality settings, choose a different port to access the stream on and add a credentials check if you don't want these URLs to be publicly accessible. Because it duplicates much of what's shown on the main *MotionEye* interface, flick the main switch to the left of Video Streaming to Off to disable it.

MotionEye can also be used to capture screenshots and record movies. Both sections enable you to choose a file-naming convention, set image or video quality, and choose how long to store these recordings for. Crucially, they give you complete control over when snapshots are taken. There's a manual option whereby a button is placed over the video feed for capturing on demand, but you can also take snapshots at set intervals, every few frames and – for security purposes – whenever the camera detects motion, as the step-by-step guide (above) reveals.

Once you've configured motion detection, you'll want to do more than simply record the event and review it at a later date when it's too late to do anything proactive about it. This is where the Motion Notifications section comes in. In here you'll find options for calling webhooks, running commands and – crucially – sending an email. Flick this switch to On, fill in the email address you want alerts sent to, then configure an SMTP server to send your emails from. You can also attach one or more photographs using the Attached Pictures Time Span field to set a figure other than 0.

The final section – Working Schedule – enables you to limit *MotionEye*'s motion detection tools to specific times of the day, ensuring you're not subject to a barrage of false-positive alerts from legitimate visitors. You can set specific hours and specific days, but remember to flick Working Schedule to Off when

you're away for a longer period, so *MotionEye* can keep an eye out 24-7 until your return.

Monitoring your home

Your camera feeds are now ready for access through any web browser or – if you're an Android user – try the free *MotionEye* app from FutureJJ.

As revealed in the annotation (opposite), clicking on any video feed reveals four buttons – this is where you go to review any snapshots or video recordings

Congratulations – you've just made your home more secure, without paying a penny in subscription fees or trusting your surveillance data to a third-party.

QUICK TIP

Be aware that GDPR covers recording of people even in public spaces, ensure footage will be deleted after a suitable amount of time. Often two weeks is suggested.

» CONFIGURE REMOTE ACCESS

MotionEye's web interface means you can keep an eye on your home at any time you like, but as things stand, you can only access it inside your own network. If you can't dial in via VPN, you need to pair MotionEye with a suitable domain or subdomain and reverse proxy. Our choice of the latter is Nginx Proxy Manager, as featured in LXF306 (see https://bit.ly/lxf306nginx).

Once you've set up your dynamic domain or subdomain, log into Nginx's browser-based interface and select Proxy Hosts followed by Add Proxy Host. Enter your domain or subdomain name into the Domain Names field and click Add, leave HTTP selected under Scheme and enter your server's IP address into the Forward Hostname/IP field. Enter port 8765 (or whichever port you configured) into the Forward Port box, then switch to the SSL tab. Click on None and select Request A New SSL Certificate (With LetsEncrypt), flick the Force SSL switch on, enter your email address, agree to the LetsEncrypt terms, and finally click Save.

When the redirect is set up, open a new browser tab and attempt to connect directly to the subdomain or domain you just configured – if all is working, you'll find yourself at the *MotionEye* login screen.

>> KEEP AN EYE ON US EACH MONTH Subscribe now at http://bit.ly/LinuxFormat

Part One! Don't miss next issue. subscribe on

FALSTAD

Emulate an analogue computer digitally

Today, computers are nearly always digital, but analogue computers also had their place, as Mike Bedford discovers through emulation.



is interested in both software and hardware, so recreating analogue computers is an ideal excursion

for him.

rom the pioneering machines of the '40s to the home computers of the '80s, we've put some fascinating computers through their paces via emulation in LXF. But they were all digital, and this ignores an important part of computing's heritage. The electronic analogue computer was important for scientific applications until the early '80s. To give these intriguing computers the recognition they deserve, we're turning to emulation to reveal their secrets.

In this first of two articles, we emulate a very simple machine, then move on to a more sophisticated one next month. We're not looking at a specific analogue computer, though - we're emulating a generic one, as models didn't differ as much as in the digital world. So, forget about bits and logic gates, and get your hands dirty with voltages and operational amplifiers.

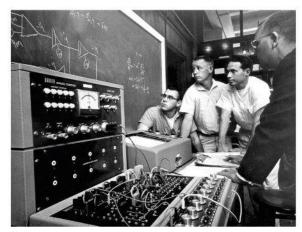


Digital and analogue computers differ in two main respects. First, the former represents values that differ in discrete steps, while the latter's values vary continuously. Second, in digital computers, changes only occur at specific times related to the clock, but in analogue computers, changes happen continuously.

These aren't the only differences, though, and most importantly, the concept of programming is very different. There's no such thing as an instruction,

> so a program isn't a sequential list of instructions. Instead, it's a definition of how the analogue computer's functional units are configured and connected together. Physically, this takes the form of adjusting the value of potentiometers and plugging patch leads into a patch panel.

> Analogue and digital computers coexisted for several decades, and we might wonder why, since history rather suggests that digital technology won because it was superior. The answer is tied up with analogue computer applications. Unlike digital computers, analogue machines aren't universal they can't solve all computable problems. Their particular niche - albeit



In contrast to the large analogue computers used in industry, desktop machines, like this one from the '60s, were used in education.

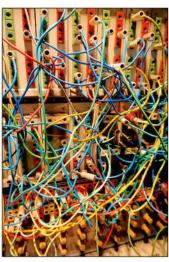
a very important one - is solving sets of differential equations, as required for simulation exercises in science, maths and engineering. On digital computers, this is computationally intensive, as evidenced by the hugely expensive supercomputers used for this sort of application today.

With the digital computers of the '50s, '60s and '70s, this could be a show-stopper, but analogue computers were much quicker. What's more, the time taken to solve a problem doesn't increase with the number of equations, a far cry from the situation with digital computers. It wasn't all one way, though, because analogue computers had drawbacks, too. Programming was often more time consuming, and accuracy was limited because of electrical noise. As a result, digital computers gained the upper hand as their performance improved, even though analogue computers clung on until the early '80s for the most demanding applications.

DIY emulator

We couldn't find an analogue computer emulator, so created our own. We built it using the Falstad circuit simulator (www.falstad.com/circuit) which can be run either in your browser or locally. You can follow our lead in creating the emulator, and this is a useful exercise because it enables you to understand the underlying

We're in two minds as to whether debugging a patching error is more difficult than spotting a typo in digital code, but it certainly requires a



www.linuxformat.com 60 **LXF311** February 2024

Analogue computer **TUTORIALS**

electronics. However, if you prefer to miss out this step and move straight to trying it out, you can download the emulator (Analogue Computer.txt), ready to use in Falstad, from https://bit.ly/lxf311code. We don't have space to explain how to use Falstad, although we found it intuitive and were soon up to speed.

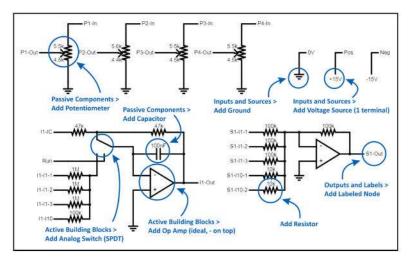
Our emulator is of a simple analogue computer that contains three types of functional unit: potentiometers, summers and integrators. A potentiometer is like the rotary volume control on old analogue radios. Its output is equal to its input multiplied by a constant that varies from 0 (fully anticlockwise) to 1 (fully clockwise). A summer's output is the sum of its inputs, although because of a quirk of its circuitry, that output is inverted, so inputs of 1, 2 and 3 give an output of -6. And an integrator's output is the integral of the sum of its inputs, again inverted.

You may recall that the integral of a constant is that constant multiplied by the variable of integration, which is time in the case of an analogue computer. So, if an integrator starts with an output of zero and a value of one is applied to one of its inputs, the output starts to ramp down. If a value of two is applied, it ramps down at twice the speed. Like a real analogue computer, our emulator has several of each of these functional units, which are connected to something that looks similar to an analogue computer's patch panel. Programming it is achieved by plugging in pseudo patch leads and adjusting the potentiometers.

The schematics for the three types of functional units, as they appear in *Falstad*, are shown above-right. Creating one of each involves dragging components on to the canvas, setting their values if they differ from the defaults, and wiring them up. The blue text shows you where to find each component in the Draw menu.

The potentiometer: This is just a single component – a potentiometer, with its bottom end connected to ground, OV. Note that the text "P1-Out" in the schematic is a labelled node, which you can find under Outputs And Labels. Note also that, unlike the other components, the value of a potentiometer isn't shown in *Falstad*. Instead, it shows the resistance of both parts, the values of which depend on its setting. However, the actual value should be set to 10k.

The summer: Built around an operational amplifier, or op-amp, which were built out of thermionic valves/



tubes in the early days of analogue computers, but are now available as chips. The gain (degree of amplification) is defined as the ratio of the feedback resistor – the resistor between the input and the output – to the input resistor. Like real analogue computers, ours has a mix of inputs with multiplication factors of one (for example, input S1-I1-1) and 10 (for example, S1-I10-1).

The integrator: Similar to a summer but with a capacitor instead of a resistor in its feedback loop. The speed of integration is inversely proportional to the value of the feedback capacitor multiplied by that of the input resistor. The integrator is complicated by the fact that its output varies with time, so it's necessary to define its initial condition. This takes the value of the input called I1-IC when the analogue switch is in its IC position, as opposed to the Run position, which is controlled by a mode switch that we'll see later.

Op-amps: These are also connected to a positive and a negative power supply, the values of which dictate the range of possible output voltages. These don't appear in the schematics because *Falstad* makes connections to +15V and -15V automatically, so the op-amp outputs can only be in the range -15V to +15V. Having created a single potentiometer, summer and integrator, duplicate them so the emulator has four of each. You need to rename the labelled nodes appropriately – S1-I1-1 would become S2-I1-1, S3-I1-1 and S4-I1-1 in the other summers. Finally in the

How our four potentiometers, summers and integrators should look in Falstad. The blue text shows where to find the component in the Draw menu.

QUICK TIP

Our subject here is electronic analogue computers, but there have also been non-electronic machines. Included in this category were mechanical devices, and even the **1949 MONIAC** computer that modelled the **UK** economy using the flow of liquids between tanks.

>> SPOTLIGHT ON SIZE

We're used to the idea of digital computers varying in size from a laptop to a supercomputer, and analogue computers also varied in size. Desktop machines were aimed at the educational market, while ones occupying several large racks were used for more demanding applications. However, while the size of a digital computer has some a bearing on its speed of operation, things are

quite different in the analogue domain.

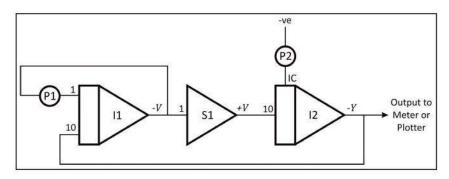
Analogue computers aren't universal computers but, even within their niche of simulation, no one machine is capable of solving any possible set of differential equations. For a start, the simplest of machines, like our emulator, didn't have multipliers, and this limited what they could do. Other functional units that might

not be available on small analogue computers include diode function generators and even some simple logic elements. However, there's a more fundamental issue. Any analogue computer is limited by its number of functional units and the characteristics of those units – for example, how many inputs a summer or integrator has. So, the four integrators in our emulator allow it to solve, at most, four

first-order differential equations, two second-order differential equations, two first-order and a second-order differential equation, and so on. But this isn't guaranteed as it also depends on how many potentiometers and summers are required. A large analogue computer – containing more functional units – could solve more complicated problems, but it wouldn't be any faster.

>>

TUTORIALS Analogue computer



The wiring diagram for our first exercise – damped harmonic motion – for you to patch up on the emulator's patch panel.

QUICK TIP

Analogue computers might not take long to perform their computations, but patching up a complicated problem could take some time. So, to make better use of the hardware. analogue computers often had multiple removable patch panels, so severa programmers could prepare their patchups in parallel without tying up the computer.

diagram, we've defined three labelled nodes named 0V, Pos and Neg, which we'll need later.

The patch panel: See an example schematic to the bottom-right. Since the functional units shouldn't be edited during programming, we suggest you place the patch panel some distance away to reduce the risk of accidentally making a change to the behind-the-scenes circuits. In the main, the patch panel comprises boxes with labels (both under Draw > Output And Labels), mimicking the appearance of a real analogue computer, with labelled nodes in each box that match up with the labelled nodes in the functional units.

The labelled nodes are the equivalent of the sockets on the patch panel. Also on the patch panel is a voltmeter with a named box, which can be connected, via a patch lead, to any other box on the patch panel, and the mode switch. We suggest arranging the patch panel similarly to ours – if you put the boxes too close together, it'll be very difficult to select a previously drawn patch lead as needed, for example, to delete it.

The mode switch: This is a physical switch and is connected via labelled nodes to the electronic switches in each integrator. To switch between its IC and Run positions, just click on the switch.

The output: A couple of things that don't appear in our circuit, which would have done on a real analogue computer, are the potentiometers and a means of seeing the value of one variable plotted against time (an oscilloscope or analogue pen plotter, for example), but both are provided by *Falstad*. Potentiometers appear in the control panel to the right of the circuit. And a time plot can be displayed by right-clicking on a socket on the patch panel, then selecting one of the scope options from the menu. By default, the scope

displays voltage and current, but we're only interested in voltage, so right-click on the scope, select Properties > Plots, and deselect Show Current.

Before using the emulator to solve a real problem, we suggest you do some simple tests to make sure there are no errors in your circuit. This involves trying all the inputs and outputs of each functional block. For the potentiometers, connect the input to +ve and adjust the potentiometer slider while you view the output on a scope. For the summers, connect the

inputs of four potentiometers to +ve and their outputs to inputs of a summer, and monitor the summer's output as you adjust the potentiometers.

There aren't enough potentiometers for all the inputs, so you'll have to rewire it to check them all. Use a similar approach for the integrators but, this time, you also need to check out the initial condition. Check the initial condition with the mode switch in the IC position, and the normal inputs with it switched to Run. Unlike the case with the summer, though, you're not interested in the output value, but its rate of change.

Plug and play

With your emulator created and checked out, it's time to use it to do some analogue computing. The power of an analogue computer is in its ability to integrate, as required to solve differential equations, so our example exercise is to solve the following pair of differential equations that define the vertical position of a weight on a spring, assuming that the oscillations will diminish due to friction. In these equations, V is the velocity of the weight, Y is its vertical position and f is a constant that is related to the amount of friction.

$$\frac{dV}{dt} = -fV - Y$$

$$\frac{dY}{dt} = V$$

As a reminder, dx/dt is the rate of change of x with respect to time, and integrating this gives a value for x.

Analogue computer programming doesn't only involve figuring out, in general terms, what to connect to what, but also carrying out scaling. This includes both amplitude scaling – that is making sure that no

>> HYBRID COMPUTERS

We've painted a picture of analogue and digital computers as being totally unconnected, but this wasn't always the case. An analogue computer can be connected to a digital computer, and the combination is referred to as a hybrid computer. These first appeared in the early '60s and continued to be used until pure digital computers became fast enough to displace analogue computing.

The analogue computer part generally carried out the simulation by solving differential equations, while the digital computer provided overall control. This was possible by permitting the digital computer to read the outputs of the summers and integrators using analogue-to-digital converters, to adjust potentiometer values, and to switch the operational mode of the analogue computer –

for example, between Run, Hold (not included in our emulator, this mode stopped integrators integrating, thus halting the computation) and IC. A classic example is an optimisation problem. Let's say we have a mathematical model of the trajectory of a projectile, and we want to adjust the angle of launch until we achieve a particular range. This is a trial and error process, but doing it manually

on a pure analogue computer could be a laborious and time-consuming sequence of setting the angle, running the simulation, observing the result, resetting the angle, running another simulation, and so on. All this changes with a hybrid computer. Although it would do the same sort of thing, it would do some more quickly by eliminating the need for human intervention.

Analogue computer **TUTORIALS**

variables exceed the computer's dynamic range (they remain within the range -15V to +15V in our case) – and time scaling. However, scaling is a major topic, which we'll conveniently gloss over and present the wiring diagram, which appears opposite and which will solve our problem. Actually, it gives us a negated value for Y, rather than a true value of Y, but it's easy to get the true value by using another summer that, if only one of its gain 1 inputs is used, acts as an inverter. If you do make that change, though, be sure you still use the –Y value in the feedback loop to I1, though, otherwise you'll end up solving a different problem.

To interpret the diagram you need to know that the circles are potentiometers, the triangles are summers, and the triangles with rectangles on their left sides are integrators. The numbers against the inputs of summers and integrators are the gains of those inputs, and while we've labelled them all, gains of one are usually omitted. We've also labelled the functional units with the numbers of the particular units as used in the screenshot (I1 for Integrator 1, P2 for Potentiometer 2 and so on), and shown the variables at the outputs of the summers and integrators. You'll notice that the symbol we're using here for a summer is the same as the one we've used previously for an op-amp. Although a summer is built from an op-amp, the two are not the same, even though electronics engineers use the same symbol for an op-amp that analogue computer scientists use for a summer.

Program the circuit in the emulator by wiring it up on the patch panel. As a general comment on patching up a problem, note that if you wire up two connections in the same horizontal row or the same vertical column, it can be difficult to differentiate the two leads, and this could make debugging unnecessarily difficult. To prevent this, we suggest that you bend some of your patch leads in this case. *Falstad* doesn't allow you to bend wires in curves, but you can use a two-segment wire. These are created as two separate wires, starting the second at the end point of the first.

Once the example has been wired up, start Falstad by clicking on the red RUN/Stop button, unless it appears as a grey Run/STOP button, in which case it's already running. Observe the output of I2 (-Y) in a scope. Ensure the emulator is in its IC mode, adjust the two potentiometers and wait until the voltage has stabilised at the initial condition you set. Potentiometer 1 is the friction-related contact f and Potentiometer 2 is the initial vertical position of the spring.

To start, we suggest 50% for Potentiometer 1 and 100% for Potentiometer 2. Now, switch the mode to Run and observe the value of –Y on the scope. You should see a curve representing damped harmonic motion, which looks like a sine wave that reduces in amplitude over time. After about 20 cycles, the position will be bouncing around zero, but that was with a fraction coefficient at 50%, so there's scope to try greater or smaller amounts of damping. The screenshot, right, shows our analogue computer emulator solving the problem.

Wiring up our solution to the damped harmonic motion problem is a good start, but to better

understand analogue computing, how about figuring out the patching yourself, just from a set of differential equations? Our problem concerns radioactive decay involving three substances. Substance A is radioactive, it has a half life related to a, and decays to produce substance B. Substance B is also radioactive, it has a half life related to b, and it decays to produce substance C. Substance C, isn't radioactive – in other words, it's stable. The following differential equations relate to the rate of change of the amount of the three substances – the three variables A, B and C.

$$\frac{dA}{dt} = -aA$$

$$\frac{dB}{dt} = aA - bB$$

$$\frac{dC}{dT} = bB$$

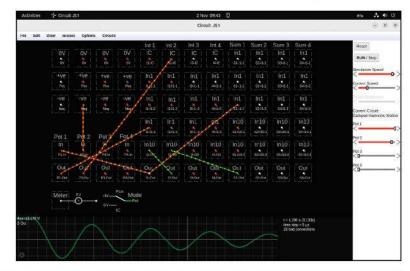
As a bit of guidance, you need three integrators, one for each differential equation, and two potentiometers, one for each of the half life related constants. These constants are actually proportional to the reciprocal of the half life. Use -ve (the minimum amount our emulator can handle) as the initial condition of the integrator for A, so it initially outputs the maximum positive value. You don't need to define the initial amounts of the other substances because they're zero if you just leave the respective integrators' IC inputs unconnected. You also need some summers, which are used as inverters. We suggest you plot all three variables on the same scope so you can see them simultaneously. You should see the amount of A fall to zero from its initial value, while the amount of C does the converse. And the amount of B should increase from zero to a maximum before falling back to zero.

Hopefully you've now got a basic grounding in analogue computing, but we're not done yet. Next month, we're going to look at an alternative way of emulating an analogue computer, one that doesn't go down to the level of electronic components but, on the other side of the coin, allows us more easily to use other functional blocks like multipliers. We'll then use that approach to solve some interesting exercises.

QUICK TIP

You might have assumed that analogue machines came first. Reality is quite different in that the two technologies were developed almost in parallel. The first electronic stored-program computer ran its first program in 1948, while the Reeves Electronic Analog Computer (REAC), which some consider the first generalpurpose electronic analogue computer. predated it by just a year.

This is the sort of thing you should see when you've patched up the damped harmonic motion circuit on the emulator's patch panel and switched it to Run.



>> TURN YOUR LINUX SKILLS UP TO 11! Subscribe now at http://bit.ly/LinuxFormat

BACK ISSUES >>> MISSED ONE?

ISSUE 310 January 2024

Product code: LXFDB0310



In the magazine

Learn Linux

with our in-depth guide to getting started with Elementary OS, or explore a more security-centric distro as we dive into Tails. Plus, find out how to perfect the art of OCR, finish your WordPress website, recreate the BBC Micro, and make your own point-and-click adventure. And no issue of Linux Format would be complete without news, reviews, Pi projects and much more besides.

ISSUE 309 December 2023

Product code: LXFDB0309



In the magazine

Manage your

apps by harnessing the power of Podman, and discover how random numbers are used by the kernel to keep your data safe. Plus, find out how to master printing in Linux, customise your WordPress website, tweak your kernel scheduler, use Python to scrape the web, process vour smart-home data, and much more. And make sure you read our Roundup of Raspberry Pi killers...

ISSUE 308 November 2023

Product code: LXFDB0308



In the magazine Bring together

all your movie, TV, music, audiobook and photo streaming needs under one roof. Plus, we take an in-depth look at the Pop!_OS distro, explain how networking works, and guide you through tutorials on WordPress, fast file management, reviving the BBC Domesday Project and more. And don't forget to check out our packed reviews section, plus we examine the brand-new Raspberry Pi 5.

ISSUE 307 October 2023

Product code: LXFDB0307



In the magazine

Power up your

Pi with awesome AI projects, and take control of your phone with the privacy-focused /e/OS. Plus, find out which open source personal cloud server is best for you, read all about the forthcoming Wi-Fi 7, and discover how to transform images into GIFs via the terminal, manage a VPS, boost your Steam Deck, and much more. And there's our usual news, reviews and sysadmin tips.

ISSUE 306 September 2023

Product code: LXFDB0306



In the magazine

Discover how

Al can (and can't) help with your coding, plus add Pi-style GPIO to your PC, access your home server from anywhere, and write your own text adventure and Breakout-style games. Also, Adminsteria returns with news and reviews for sysadmin types, we take an in-depth look at how Azure stacks up against AWS, there's a Roundup of mediacreation distros, and plenty more...

ISSUE 305 Summer 2023

Product code: LXFDB0305



In the magazine

Don't let Big

Tech be in control of your documents - find out how to run your own office apps in the cloud. Plus, learn how Linux certification can kick-start your career, step back in time with our look at the evolution of coding, and read our recommendations for the best open source CAD packages, not forgetting our usual packed tutorial, review and Raspberry Pi sections.

To order, visit www.magazinesdirect.com

Select Single Issues from the tab menu, then select Linux Format.

Or call the back issues hotline on 0330 333 1113 or +44 (0)330 333 1113 for overseas orders.

Quote the product code shown above and have your credit or debit card details ready.

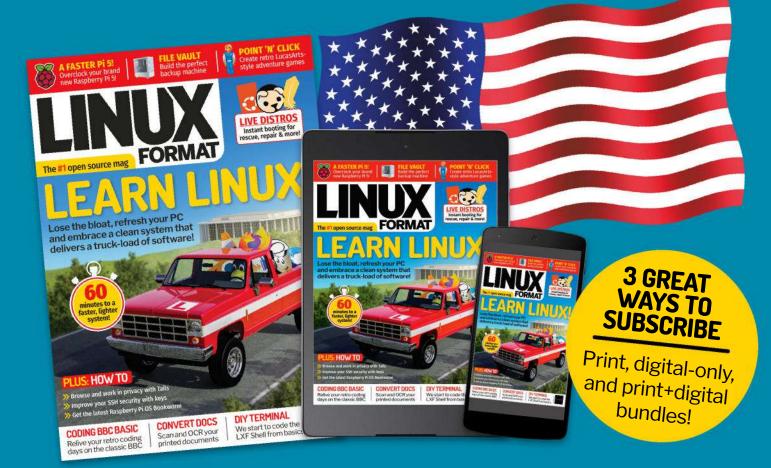
64 **LXF311** February 2024 www.linuxformat.com

USA? EU? THE MOON?

UK readers turn to p16

SUBSCRIBE!

Don't wait for the latest issue to reach your local store – subscribe today and let *Linux Format* fly straight to you. Faster, cheaper and with DRM-free archive access!



SUSAFrom \$125.99
For 13 issues

>>> REST OF THE WORLD From \$125.99
For 13 issues

>> EUROPEFrom €110.99
For 13 issues

IT'S EASY TO SUBSCRIBE!

Visit www.magazinesdirect.com/linux-format Call +44 0330 333 1113

Lines open Monday-Friday, 9am-5pm, UK time.

*We don't actually deliver to the Moon. Yet.

ADVENTURE GAME STUDIO

Credit: www.adventuregamestudio.co.uk



Create and use inventory objects

Nate Drake invites you to relive the glory days of point-and-click adventure games by creating your very own.



EXPERT
Nate Drake

Nate Drake is a tech journalist specialising in cybersecurity and retro games. The first thing he did when discovering Linux in 2004 was play Beneath a Steel Sky. Be vigilant. n part one of this series, we indulged in the nostalgia of playing classic point-and-click games from the '80s and '90s, such as *Monkey Island, Broken Sword* and *Beneath a Steel Sky.* We also covered setting up one of the best and simplest-to-use game editors for the genre, *Adventure Game Studio (AGS),* which can create games to be run on Linux.

AGS has excellent documentation and sample files, which we used to set up your first room and define walkable areas for the character. We also discovered how to have the character walk behind objects, as well as set clearly defined edges for the room, so the game knows when to load new areas.

The final part of the tutorial involved creating hotspots These are fixed parts of the background, such as statues and trees, with which the player can interact. If you've followed us this far, there are now two hotspots in your first room: a glowing orb and a spaceship, which the character can examine by right-clicking.

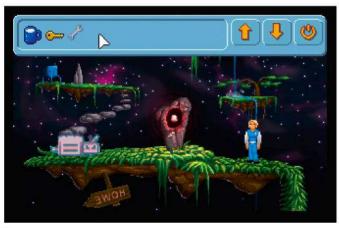
By default, the player can interact with a hotspot from anywhere, but if you want a more *Monkey Island*style game where characters walk up to objects first, follow the steps in Walk-To Points (*opposite*) before continuing.

Importing objects

The essence of any point-and-click adventure is picking up certain items and using them in context. Classic games usually make you jump through various hoops to obtain what you need, such as in *Day of the Tentacle*, where you have to place a blanket in a chimney in order to pick up a gold-plated quill pen, which is one of the items you need to trade with an NPC (non-playable character) to obtain a battery.

AGS refers to such items as objects – anything in the room that can disappear or needs to be animated.

While daisy-chaining objects LucasArts-style will come in time, for now we're going to keep things very simple by introducing one object to Room 1 that your character can inspect and pick up. To get started, click Sprites in the Project Tree, then Main in the folder tree,



Move the mouse to the top of the play window to view the inventory. Right-click to inspect objects, left-click to use them.

top-left. You'll see an image already in here. This is a blue cup, which is already in your character's inventory.

Click Import New Sprite(s) From Files. From here you can add a new image. The AGS documentation recommends doing this with a key, but as (by its own admission) this is boring, we're going to add a wrench.

If you don't want to grapple with creating your own pixel art just yet, you can download a sample image from: https://github.com/azuregate/pointandclicklxf/blob/main/wrench.png.

This opens the Import Sprite window. There are tools here to make importing images easier, such as tiling and setting transparency. But if you're using our sample wrench, it's already the right size and has a transparent background, so go ahead and click Import.

Add your objects

Now your wrench is imported, it's time to add it to the room. Go to Rooms > Edit Room in your Project Tree.

From here, go to the Room Parts section and choose Edit This Room's > Objects. Move your mouse to where you want to place the object and right-click. Select Place New Object Here.

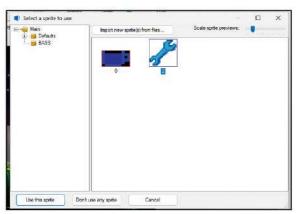
You'll now see an image of a blue cup. This is the default sprite AGS displays whenever you place an undefined object. You'll also see a new Properties box has appeared at the bottom-right. Click the ... on

QUICK TIP

Stay on top of the latest code, find it here: https:// github.com/ azuregate/ pointand clicklxf/blob/ main/rooml.asc

66 **LXF311** February 2024

Point-and-click adventure TUTORIALS



By default, new objects appear as a blue cup. Click Image to update them to another sprite you've imported.

Image to open the Select A Sprite To Use window. Click the wrench, then on Use This Sprite.

Next, in the Properties grid, scroll down and give the object a plausible Description and Name, such as Wrench and oWrench. Read through the other properties here; the most important ones are Clickable and Visible. These are set to True by default and for now, we'll leave them as is. Still, most point-and-click games make you perform certain tasks in order to interact with or see some objects, so bear this in mind if you plan to create a fiendishly difficult adventure.

Adding to inventory

Before doing anything else, save and run the game by pressing F5. At this stage, if you try to pick up the wrench by left-clicking on it, your character says he doesn't need it. If you examine the wrench by right-clicking, the character says "Looks Alright".

You can set a more detailed description in the same way as you did for hotspots. Make sure your chosen object is selected, then in the Properties grid, click the Events (lightning bolt) icon. Click the ... besides Look At Object, then add your description, for example:

Display("A solid steel wrench. It might come in handy.");

Next, expand Inventory Items in the Project Tree at the top-right. You'll see there are two entries (iBlueCup and iKey) for the two default items your character has in his inventory.

Right-click Inventory Items and choose New Inventory Item. First look at the Properties Grid and set Description and Name to something meaningful for this inventory item, such as Wrench and iWrench.

Note that the property PlayerStartsWithItem is set to False. In this case, this is what we want, as the wrench is lying on the ground when the game begins. However, you can set this to True for any objects you want to have already in the character's inventory.

The main window shows the blue cup as the default image, so make sure to select the ... in Image to update the correct sprite to show in the inventory (in this case that of the wrench). Do the same for CursorImage.

Managing inventory objects

Take a moment to save your work, then return to the Room1 tab. Double-check your new object is selected and then return to Events in the Properties Grid.

From here, we can tell the game what to do when a player tries to pick up the object. If you chose the

BASS default theme as recommended last month, there's no specific in-game "pick up" command, so click on the ... next to Interact Object to remove the wrench from the floor and place it in the inventory:

Save your work and run the game by pressing F5. You should now be able to left-click on the wrench to have it disappear from the ground and be added to your inventory. Move your mouse to the top of the screen to see the wrench in there alongside the blue cup and key. You can also click on the wrench to use it on other objects or hotspots, though that doesn't currently do anything meaningful.

As you're playing, you may notice that when you right-click any inventory item, the character says, "It's just some junk in my inventory." You may want to add a more helpful description for your new inventory item. Quit the game and return to your chosen object (iWrench) under Inventory Items in the Project Tree.

From here, click Events in the Property Grid, then select the ... next to Look At Inventory Item. If you've come this far, this is an important step in creating your own game, as so far we've only focused on one room.

Here we're opening **GlobalScript.asc** to edit the whole game script, as naturally the description of this inventory item applies no matter which room you're in. Add something meaningful, such as:

Display("A steel wrench. It's heavy but useful.");

Save the game and run it again by pressing F5. Pick up the wrench to add it to your inventory, then right-

QUICK TIP

You can use an Al image generator like DALL-E (https://openai.com/dall-e-2) to create objects. Use the prompt "pixel art" or "32-bit pixel art" to create images that fit the game's theme.

> WALK-TO POINTS

In the previous tutorial, we covered creating and looking at hotspots, such as the glowing orb and spaceship. This is fine for examination, but as your character starts to physically interact with hotspots, it makes no sense if they do so from the other side of the room.

Many point-and-click games like *Monkey Island* have your character walk up to objects before you interact with them. Happily, *AGS* supports this, too.

To enable the feature, first go to the Project Tree and select General Settings. Next, from the main window, change Automatically Walk To Hot Spots In Look Mode to True.

Return to Room > Edit Room in the Project Tree and choose Edit This Room's > Hot Spots > (Your Chosen Hotspot) in the Room Parts section. Select the Properties button from the Properties box at the bottomright. Here you can set the coordinates to which you want your character to walk when interacting with the hotspot.



Use Mouse Coords to set WalkToPoint to somewhere within a walkable area, so the player can interact with the hotspot.

If you move your mouse into the main window, you can see the Mouse Coords helpfully displayed, so you'll know which numbers to enter. Make sure the coordinates are also inside the Walkable Areas you defined.

In the case of our testing room, we choose 141,139 for the glowing orb, which places the default character just to the left of the rock feature that holds it. Repeat this for as many hotspots as you wish.

To remove a walk-to point, just set WalkToPoint to 0,0.



TUTORIALS Point-and-click adventure

OUICK TIP

Once you've chosen to add a new object image, you aren't stuck with that particular position. Just use the mouse to click and drag it elsewhere in the background. Remember to keep it within walkable areas.

click to view your new inventory item description. Remember above all that inventory items are not the same things as objects. Picking up an object may give you an inventory item but these items stay with a character throughout the game unless they consume or drop them. Inventories are also unique to each character, so if you create a game for two players or more, they each have their own items.

Walk up to pick up

If you took the time to give your game a test run, you'll have noticed that your character can pick up the wrench from anywhere – even from the other side of the screen.

Although it's possible to set specific WalkTo coordinates in the Properties Grid for hotspots, you need to put in a little more effort if you want players to walk up to items before picking them up.

Return to the Room 1 tab and hover your mouse over the object. Make a note of its Mouse Coords. In the case of the wrench, this is 246,148 for us.

Next, make sure your object, oWrench, is selected in the Room Parts section. Look to the Properties Grid and select Events. From here you can double-click to open the Interact Object code once again.

As things stand, the object disappears as soon as the player interacts with it. Modify the code to read: player.Walk(246,148,eBlock);

>> THE IMPORT SPRITE WINDOW

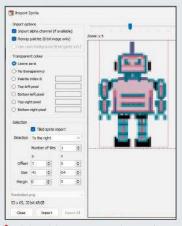
You can import images simply by clicking Sprites from the Project Tree, then choosing Import New Sprite(s) From Files, top-left.

If you choose an image, you'll see the Import Sprite window. This shows your chosen image with various options. Use the Zoom slider at the top to see it more clearly.

The easiest way to import images is to make sure they already have a transparent background. We recommend using the fuzzy select tool in GIMP (https://gimp.org) to do this, as it's simple to use and free of charge. Once done, simply leave the default Import Alpha Channel option ticked and then hit Import or Import All in the case of multiple images.

If your image has a set background colour, you can easily render this transparent in the Transparent Colour section by choosing one of the corner pixels – Top Right, for example.

If you only want to import a section of an image, click and drag with your mouse – only the area highlighted in pink is used.



Click and drag to import only part of an image as a sprite. You can also use this to import a grid of tiled images.

The Tiled Sprite Import feature is useful when you're trying to import a series of images – for a character who's walking around, for instance. It enables you to import a grid of sprites into separate slots.

Just tick the Tiled Sprite Import box, and align your rectangle on the top-left sprite in the series. When you click the left mouse button, you get an extra step, which enables you to size the grid.



Players can click on an inventory item to use it on another object. The cursor changes to reflect the active item.

oWrench.Visible = false; player.AddInventory(iWrench);

This new line is quite easy to follow. The player.

Walk command tells the game to have the character walk to the specified coordinates for the object, which we noted earlier. The eBlock parameter is crucial, as it tells the game to wait until your character has arrived at that spot before executing the next lines of code to make the item disappear and add it to your inventory.

Using inventory items

By default, the game allows you to interact with inventory items just by clicking on them. This can make sense in the case of consumable items, such as the blue cup. Indeed, if you run the game and left-click on the cup in your inventory, you'll see the character saying that they don't think drinking it is wise.

Take a moment to double click iCup under Inventory Items in the Project Tree. Click on the ... besides Interact Inventory Item in the Properties Grid to open the global script to view the default message. Feel free to update the description if you want – for example, "You drink from the cup. It's now empty."

Most often in point-and-click adventures, though, it's a matter of using the right item at the right time. As you saw earlier, when you click on the key or wrench inventory items, the cursor changes to resemble them. This indicates that you can use these inventory items on a hotspot or object to trigger an in-game event.

To keep things simple, let's first have the game display a message if you try to use the wrench on the spaceship hotspot.

First return to the Room1 tab and in the Room Parts Section, choose Room > Hotspots > hSpaceship.

Click the Events button in the Properties Grid and scroll down to Use Inventory On Hotspot. Clicking on the ... here opens the global script once again. Here we'll define a function to display text when you try to use the wrench on the ship:

function hSpaceship_UseInv()
{
 if (player.ActiveInventory == iWrench) {
 Display("You can't use the wrench on the ship. It's too
far away.");
 }
}

As you can see, AGS scripting is very easy to follow. First the code checks if the active inventory item being

68 LXF311 February 2024 www.linuxformat.com

Point-and-click adventure TUTORIALS

used by the player (**player.ActiveInventory**) is the wrench (**iWrench**). If so, the message displays.

Save and run the game once again. First go to your inventory and click the key to change your cursor to the key icon. Notice that nothing happens if you try to use it on the ship. Next pick up the wrench, select it from your inventory and then click the spaceship. You'll see the message only displays when you're trying to use this specific inventory item.

Robot repair

So far you've learned how to use an inventory item to display a message, but point-and-click adventures tend to be much more interactive.

In order to get used to the mechanics of this, we're going to introduce a very simple puzzle to your room. On the left-hand side of the room, there will be a pile of robot parts. If the player picks up the wrench and uses it on the parts, they will have assembled their very own robot.

To get started, you need two images for these objects. One is of the broken robot parts and the other is of the working robot. You can download samples of both of these from https://github.com/azuregate/pointandclicklxf, though Nate strongly recommends you rely on your own artwork over his.

Once the two files are downloaded to your computer, open AGS and click into Sprites in the Project Tree. Choose Import New Sprites From Files at the top-left and select both **brokenrobot.png** and **fixedrobot.png**. Choose Import All to add the sprites.

Next, return to your Room1 tab (or click Edit Room in the Project Tree). In the Room Parts section, choose Objects and then right-click to select Place New Object Here. As before, the blue cup appears, so click Image in the Properties Grid to update to the image of the broken robot.

Update Description to Robot Parts and give this object a useful name, such as oBrokenRobot.

Right-click the same area again to add another object. Update the image for this to that of the fixed robot. Add a meaningful description, such as Fixed Robot, and name, such as oFixedRobot. Under Visible, choose False, as this sprite should only appear once you've assembled the robot using the wrench.

Next, return to the Room Parts section and make sure that oBrokenRobot is selected. In the Properties Grid, click the Events button, then choose the ... next to Use Inventory On Object. Add the following code:

function oBrokenRobot_UseInv()
{
 if (player.ActiveInventory == iWrench) {
 player.Walk(86,140,eBlock);
 oBrokenRobot.Visible = false;
 oFixedRobot.Visible = true;
 }
}

As you can see, this code once again checks that the active inventory item – the cursor image – is the wrench. If so, it instructs the player to walk a little to the right of the robot parts. The broken robot image is then made invisible in favour of the fixed robot.

Although it's not essential, you may want to update the description for the fixed robot to remind the player of what they've done. Make sure oFixedRobot is selected in the Room Parts section. Click Events, then the ... next to Look at Object to add a useful description – for example:

Display("You fixed the robot. It seems to be working.");

Take a moment now to save and run the game using F5. As you'll see, only the robot parts are displayed. Click to pick up the wrench, then click it again in your inventory to use on the parts to fix the robot.

Trash talk

Some point-and-click adventures place an upper limit on the number of items you can have in your inventory. If there are items your player can only use once, it pays not to hang on to them, as it can cause confusion.

For the purposes of our game, let's assume that the wrench is a one-use item. Return to the event Use Inventory On Object for oBrokenRobot and add the following lines:

Display("As you're fixing the robot's arm it snaps the wrench in two. It must be ticklish."); player.LoseInventory(iWrench);

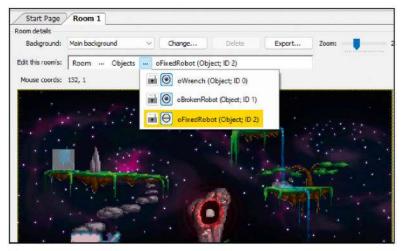
The **Display** command isn't absolutely necessary here but it's best to give your player an explanation of why their inventory item has disappeared. You can use the **LoseInventory** command to remove any inventory items.

Stay tuned!

If you've followed the steps in this guide, by now you should be comfortable with creating objects that your character can examine and pick up. You should also be able to inspect and use inventory items on both hotspots and objects in the room.

In the next part of the series, we're going to delve into how to do more with items by adding some animations, as well as work out how to manage conversations with characters.

You can download the code for Room 1, including all changes made so far, from: https://github.com/azuregate/pointandclicklxf/blob/main/room1.asc.

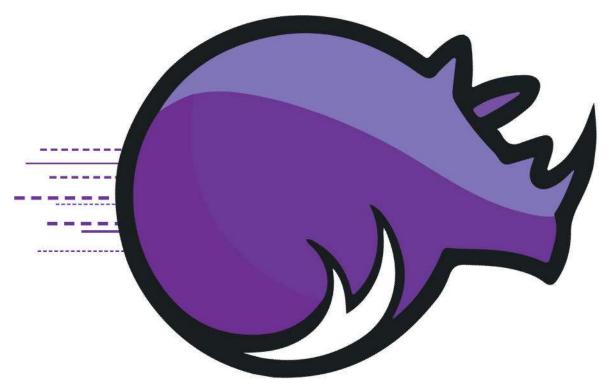


Remember, you can click the eye icon next to objects in the Room Parts section to hide or unhide them in the Room Editor.

QUICK TIP

If your game has a point system, add the code line GiveScore(X); - for example GiveScore(10); - to events to reward the player for carrying out correct actions. We'll explore point scoring in more depth later in the series.

>> ADD LXF TO YOUR INVENTORY! Subscribe now at http://bit.ly/LinuxFormat



Roll out the Rhino!

In an increasingly crowded marketplace, **Nick Peers** discovers why this new Ubuntu derivative might be a barrel of fun!

SPECS

CPU: x86/x64 or ARM64 PC, Pine or Raspberry Pi **Mem:** 2GB recommended **HDD:** 32GB minimum

here are already plenty of Ubuntu variants out there, so what makes Rhino Linux stand out? The clue is in its former name, Rolling Rhino Remix. Unlike Ubuntu and virtually all its derivatives, Rhino Linux follows a rolling release model, which means that rather than freeze its core repositories at point of release, as each major version of Ubuntu does, the packages within can be updated with the latest versions at any point to ensure you're always working with the latest available release.

The concept isn't new to Linux, of course – Arch Linux is the most obvious proponent of this approach, a single version that is continuously updated as opposed to the separate point release approach of Ubuntu. The secret to Rhino's successful adoption of this approach – like so much else in its code – is the clever integration and cannibalisation of existing tools and technologies, as we'll see in this feature.

The project first saw the light of day under the title Rolling Rhino Remix in late March 2022. At first it was little more than a series of script-based tools that handled the updates, but it soon started to investigate alternative methods.

In October 2022, Rolling Rhino Remix was shelved in favour of a more ambitious successor (see https://rollingrhino.org for details). Rhino Linux 2023.1 rolled out in August 2023 as a fully functional Ubuntu variant, with its own custom desktop and wrapper that brings together multiple package managers under one roof. Version 2023.3 rolled out in October, offering builds for both ARM64 and x86/x64 PCs as well as Pine and

Raspberry Pi devices, so we've decided now is the perfect time to take it for a test drive.

We'll step you through the (simple) installation process, take you on a complete tour of the desktop and – of course – check out its package managers to see how it delivers on its promise of continuous updates. Strap yourself in and let's roll...

Rhinos and unicorns

From the moment you spin up the Rhino Linux live disc, you can see this operating system is easy on the eye. From the early glimpse of the distro's Unicorn desktop in the live environment to the pleasing install wizard, a heavily modded version of the *Calamares* installer, giving Rhino Linux a test run or installing it on your main system or in a VM is a relative breeze.

There are some caveats: attempts to access the release note and support links from the main installer wizard throw up a 'Failed to execute default Web Browser error: Input/Output error' message – setting a default browser has no effect here. The rest of the install wizard is a pleasure to work through, with some nice touches (a graphical representation of each keyboard model and layout, for example), but our chief highlights are the partitioning options, which cover all bases, and the summary screen at the end of the wizard giving you a second look at your proposed partitioning scheme before you commit to the install.

It's worth noting that unlike Ubuntu, Rhino doesn't offer you the opportunity to preinstall non-free drivers during setup, so if you have Broadcom or Realtek Wi-Fi

Rhino Linux IN DEPTH



Rhino Linux's installer offers you a wide array of tools for setting up your hard drive prior to installing it.



Updates are, primarily at least, delivered through the Your System tool – just one click and one password entry required.

drivers, you need to connect your PC to the network via Ethernet cable or tether it to your mobile phone via USB during setup to provide the mandatory internet access required until you can download them later.

Another annoyance raises its head during the install process – among the highlights on offer are what Rhino Linux calls "elegant new setup wizard", which shows a screen allowing you to choose what package managers to support (Snap, Flatpak and Applmage) alongside the default *APT* and *Pacstall*. However, there was no sign of this wizard after installation on two separate machines – a quick scour of Rhino's Reddit forums revealed several possible fixes, but none worked. It turned out the **rhino-setup** package was missing, but a trip to its repo revealed how to install it: \$ pacstall -I rhino-setup-bin

Ironically, this gave us our first encounter with what powers Rhino's rolling release setup: *Pacstall* (https://pacstall.dev), a tool that provides rolling updates for just shy of 500 packages, and which is incorporated into *Rhino-pkg*, Rhino's package management tool. More on that shortly.

Once in place, the setup wizard can be launched by right-clicking the desktop and choosing Applications > Other > Setup Wizard. This opens with a straight choice of dark (default) or light colour schemes, then offers a choice of which package managers to install and incorporate into Rhino followed by some additional settings (the Nala front-end for *APT*, enabled by default, plus options to add GitHub CLI and the Apport crash reporting system – for developers only). You're

then prompted to enter your password several times before rebooting to complete setup.

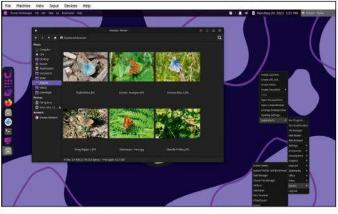
A horny desktop

You can now explore Rhino's Unicorn desktop, a custom-built variant of Xfce. This ensures it's lightweight – just 600MB RAM with all customisations enabled – but with what Rhino describes as a "modern" facelift, something we can't disagree with compared to Xfce's rather dated, if functional, look.

This is the area where Rhino's developers have lavished most attention, and for the most part it works well. Rather than reinvent the wheel, Rhino manages to combine icon, GTK and WM themes from multiple sources (Elementary's Xfce Darker, Xbuntu's Greybird and Ubuntu's Yaru Dark) to keep a level of familiarity while doing just enough to make Unicorn feel fresh.

It's understandable due to all the effort poured into it, but Unicorn is the only desktop environment in town as far as Rhino is concerned. You can, of course, switch desktops as easily as with any Ubuntu variant, but there are no officially sanctioned alternative installers. Rhino's developers hope to unveil more desktop spins, but have handed this job over to the community, with a major caveat: they have to "follow Unicorn's aesthetic, icon theme, colour scheme and components". This has resulted in the Unicorn Beyond XFCE Initiative (UBXI), which should see official – albeit unofficially maintained – UXBI desktop spins rolling out in due course.

Returning to Unicorn itself, those who've played around with Xfce 4.18 will spot familiar elements, most



Rhino's Unicorn desktop doesn't ship with a traditional apps menu - right-click the desktop to access its equivalent.



The Unicorn desktop is a hotchpotch of other tools – Xfdashboard, in the case of its desktop switcher – cleverly stitched together.

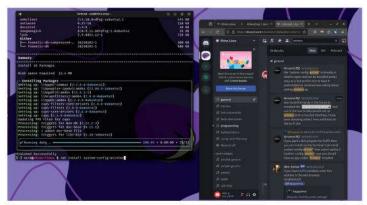
>> GET HELP AND SUPPORT

Rhino Linux is a relatively young distro, which means that it still suffers from some teething troubles. There's also a relative lack of documentation – visit the site wiki (https://rhinolinux.org/wiki.html) and you'll see that few topics are covered. However, that doesn't mean you're left high and dry – remember that the components of Rhino's Unicorn desktop have been cannibalised from elsewhere, so look for a Help option to be redirected to the relevant online documentation (such as the Xfce Panel) or make a note of the original component's name and search online for instructions.

When it comes to getting support and speaking to the project's developers, things are more promising. Head to Rhino Linux's GitHub page (https://github.com/rhino-linux), where you'll find links to Reddit (www.reddit.com/r/rhinolinux), Matrix (https://matrix.to/#/#rolling-rhino-remix:matrix.org) and Discord (https://discord.gg/uhdCz8vwV9). A peek at the Reddit site reveals some support, but you'll find the developers are most visible and active on Discord.

Here you'll find a handy search tool can quickly track down a solution to a specific problem, but beginners may struggle to fill in the blanks. For example, one user noted the apparent lack of printer support, which was quickly rectified by other users pointing him towards the Ubuntu GUI (rpk install system-config-printer). However, there was only the vaguest reference to also requiring CUPs, with no explicit instructions (rpk install cups – choose option 5 – if you're interested).

If you still can't find an answer, remember Rhino's origins – expand your search to include Ubuntu, which may help resolve some issues.



There's a thriving support hub at Discord, where you'll find fellow users and Rhino devs standing by to provide help and support.

notably the menu bar, which sits across the top of the desktop. There's a lot of customisability behind the scenes, with a rich array of panels to choose from, but in its initial state you'll see Unicorn places a power button on the left of the menu bar and a series of status bar panels (time/date, various hardware devices) on the right. To the right of the power button, you'll see context-sensitive menus based on the currently selected application or – failing that – the desktop itself. The Desktop menus provide quick access to desktop and system settings, plus expandable menus enabling you to browse user folders without having to open Unicorn's *Thunar* file manager.

Exploring the dock

Beneath the menu bar on the left – aping the approach found in Ubuntu's Unity desktop – you'll find a colourful and compact dock with access to a limited array of

shortcuts. As with most docks, you can pin more after opening the program in question (when its icon appears in the dock, right-click it and choose Keep In Dock). There's a limited selection to start with: *Firefox, Thunar, VSCodium,* the standard Xfce terminal and a shortcut to the Your System applet (see below).

Above these are three desktop tools: Search Bar, Application Grid and Desktop Switcher. The search bar is the most useful of these, based on the superswift uLauncher search tool. Click its icon or press Win+S to bring up a simple dialog, then type the name of the app you're looking for or type / or ~ to browse your filesystem.

Application Grid, which is based on Lightpad, crams too much into a single screen in our view – we prefer the more rationed experience of Gnome, which isn't afraid to spread icons over multiple pages to reduce clutter. However, Desktop Switcher is another delight – based on Xfdashboard, its full-screen approach makes it easy to manage and move between multiple desktops. It also provides access to apps and system tools through an organised hierarchy as well as a slightly gimmicky analogue display.

Returning to the desktop, you'll also find a lot of functionality packed into the desktop context menu – right-clicking reveals the usual desktop-based tools, such as creating items from scratch and rearranging desktop icons, but also unveils the Applications menu, a hierarchy of organised shortcuts usually associated with the menu bar itself. All this probably explains the lack of a proper application menu on the menu bar, but if, like us, you feel bereft without one, you can easily add in the alternate Xfce Whisker Menu launcher via the Panel > Add New Items dialog, once again highlighting the fact that this is a desktop unafraid to hand over control.

Package managers

As mentioned, one prominent shortcut on the dock takes you to the Your System applet, which provides a basic rundown of system information. Its main function, however, is revealed by System Upgrade at the bottom. Clicking this brings up a miniature terminal window inside the applet – enter your user password and Rhino Linux performs a full system update using a smartly formatted progress bar. On our test system, this included upgrading the kernel itself to 6.6.0 from the build (6.5.5) provided in the 2023.3 release, thanks to *Pacstall*.

That said, upgrading through Your System didn't deliver a comprehensive update. It turns out that you also need to periodically update Rhino's package manager – *Rhino-pkg*, or rpk as a shortcut – separately. This wraps *Pacstall*, *APT* and – if you selected them earlier – Flatpak and Snap together, providing you with a one-stop shop for all your app installation and updating needs. By using rpk in place of apt, snap or flatpak in the terminal, you're shown available packages across all supported package managers, enabling you to choose whichever one(s) you prefer.

While the syntax is logical and it's easy enough to employ, the lack of a graphical front-end is a major omission for an OS trying to sell its user-friendly credentials. We'd also like to see more useful information – specifically version numbers – to help

72 | XF311 February 2024 www.linuxformat.com

differentiate between the available releases of a program across multiple package managers.

Updating is as simple as issuing the following: \$ rpk update -y

Enter your user password and a complete update of all repos – including *APT* and *Pacstall* – is performed, confirming that Rhino ties itself to Ubuntu's devel repositories rather than a specific release. On our test machine, we were prompted to upgrade nala-deb (the superior terminal front-end for *APT* packages) to complete the update process.

Rolling updates

Ultimately, Rhino Linux will stand or fall on its promise to deliver a version of Ubuntu that offers continuous rolling updates. To achieve this, many of those updates are delivered using *Pacstall*. Inspired by *AUR* (*Arch User Repository*) in Arch Linux, this ensures that once installed on your system, you won't ever have to perform any major upgrades to Rhino itself; instead, *Pacstall* ensures you get the latest version of the Linux kernel, plus is used by Rhino's developers to deliver Rhino-specific updates, including the core Rhino themes and apps, and other essentials, such as *Firefox*.

While Pacstall is an AUR wannabe, it has a long journey to close the gap between the two. AUR contains over 85,000 packages; Pacstall currently has fewer than 500, but that number has more than doubled over the past year. It's worth browsing these (see https://pacstall.dev/packages) to check whether any key packages you need are covered, but remember that Ubuntu's devel repos are also frequently updated, unlike with regular Ubuntu releases.

Moving on, the Rhino Linux homepage lists a fourth major highlight: *RhinoDrop*. As its name implies, it's a clone of Apple's AirDrop, but its presence is a bit of a misnomer. The feature links any computer or mobile on the same network via a single web page (https://drop.rhinolinux.org) to enable you to easily shunt files between them. It's basically a rebranded fork of *Snapdrop*, so unless there are plans to integrate some of its features into Rhino Linux itself (say an option to

```
[-] nick@rhino-linux $ rpk install bitwarden
Searching flatpak...
Found packages matching 'bitwarden':

[0]: com.bitwarden.desktop (flatpak)
[1]: bitwarden-cli-bin (pacstall)
[2]: bitwarden-deb (pacstall)
[3]: bitwarden (snap)

Select which package to install [0-3]: 2

Selecting 'bitwarden-deb' from package manager 'pacstall'
Are you sure? (y/N) y
(bitwarden-deb) Do you want to view/edit the pacscript? [y/N
```

The Rhino-pkg tool searches across all available package managers to give you a choice of where to install apps from.

share files via *Thunar's* Send To menu), we're not quite sure what its purpose is in the grand scheme of things.

Final thoughts

This brings us neatly to the ultimate question: does the promise of rolling updates make Rhino Linux an irresistible sell to all Ubuntu-based users? The answer has to be no – for now. But a lot of our reluctance is down to the fact Rhino Linux is a relative newcomer.

It's clear that Rhino shows much promise – a lot of work has gone into its desktop, even if it's largely a mashup of other elements. Integrating *Pacstall* to provide the means to supply a single, continuously updated build rather than just commit to regular point updates certainly gives it a unique opportunity to find space in the crowded world of Ubuntu variants.

But there's still work to be done. The well-crafted *Rhino-pkg* provides a one-stop shop for sourcing, installing and managing apps from multiple sources, but the fact it's only accessible from the command line will put many off. At the same time, despite the sleek desktop, there's still a feel of belts and braces about some of the experience, from the malfunctioning links in the installer to the vanishing setup wizard. The inclusion of developer-friendly tools like *VSCodium* suggests that Rhino may also be trying to woo coders, but this seems like it's caught between two competing audiences. Until it evolves further and makes a firmer choice over which audience to go after, we suggest you stick to giving the distro a run in *VirtualBox* before committing to a full-blown install.

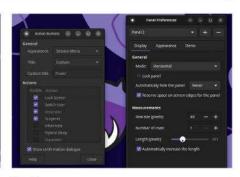
CUSTOMISE THE RHINO MENU BAR



Reconfigure the menu bar
To add elements such as a launcher
to the menu bar, right-click anywhere on it
and choose Panel > Add New Items. You'll
see a range of options, including various
application menus (we recommend the
Whisker Menu) and status tools. Click Add
to place them on the right of the panel.



Customise the bar further
Right-click an existing panel entry
and choose Properties to customise its
appearance on the bar (show icon and/
or label, plus panel-specific settings).
Choose Move to drag the item to a
different position on the bar and Remove
to delete it from the panel completely.



Add more panels
You can also add extra panels, which
can be placed anywhere on screen. To do
this, right-click the menu bar and choose
Panel > Panel Preferences, then click +
next to Panel 1. Adjust its size via the
Display tab and add new items using
the Items tab. Click Close when done.

www.techradar.com/pro/linux February 2024 LXF311 73

ADMINISTERIA



Stuart Burns is a Linux administrator for a Fortune 500 company specialising in Linux.

>> LINUX LAPTOPS

I've been growing dissatisfied with living in a Mac OS-centred world, so have swapped my £1,800 MacBook Pro for a £250 second-hand ThinkPad as an experiment. The result? It was great. Stuff that never worked on the Mac (looking at you, dualmonitor support) worked out of the box with Ubuntu on a T480.

Why do it? For one, in my opinion, Apple has been going down a route that betrays the Linux-friendly original hacker ethic: near zero repairability, the creeping of 'You must log in to use the App Store to install this app' and adverts.

Secondly, I wanted to see if it could do all the work stuff I could do on a Mac. So far, the answer is yes. Cloud computing helps, with increased support for Linux (at least in the Azure world I inhabit).

In other news, I'm hearing that, as predicted, companies are re-evaluating their RHEL environments, and more and more companies (no metric, just conversations with other admins) are moving to Ubuntu.

It does make sense, to be able to just download the ISO or do a quick deployment, rather than having to register to download the ISO and taking up precious, expensive licences.

Software vendors are not stupid. They realise that they are going to have to choose their allegiance in terms of which OS they support, and RHEL isn't looking too clever.

The RHEL ecosystem hasn't burst yet, with its expensive but well-regarded training courses, but the cracks are starting to appear where the big customers who pay the big money are moving to alternatives.

Manage network downloads locally

Find out how to apply some admin knowledge to network downloads.

e're sure we're not the only ones who have tried to download a large file while watching Netflix at the same time. The stuttering and buffering will drive you crazy. There is no simple built-in way to manage local network I/O.

The fix? Wonder Shaper. It's a cute little app that enables you to use the command line to set a rate limit (Kb/s) for the local device. Using it in such a fashion is a rough and ready way to manage I/O allocation.

To install Wonder Shaper, just use: \$ sudo apt install wondershaper

Setting up the speed limit on the device is quite simple:

\$ sudo wondershaper enp0s1 5120 1024

This would set the device enp0s1 to allow a five-megabit download and one-megabit upload. It can be a bit temperamental, but in a pinch it can it be useful. We suggest using *Wget*, so any break is resumable.

To remove the limits when you've finished watching Netflix is a trivial matter. Just use the following command and your Ethernet device will go back to the way it was before: \$ sudo wondershaper clear enp0s3

Farewell, Docker; hello, Rancher Desktop

Introducing yet another – better – way to manage your containers.

he whole *Docker* limitation for business has a real impact on developers. As an example, I am not allowed to use *Docker Desktop*, even though I manage and maintain business-critical image builds. It's a pain. I have mentioned *Podman* in the past, but I recently stumbled across *Rancher Desktop*. At its core, it's a desktop management system for

system management and testing, but it can also be exceedingly useful in the development phase, too. While it's not exactly the same as *Docker*, it is certainly well worth checking out.

It just makes life easier by allowing easy, same-as production, using production-like commands to build and test. Best of all for those in big business, there is no bill to pay

per month to use Docker.

At the same time, the project is sponsored and run by SuSe, the Linux vendor, so there is a good degree of stability and less worry knowing that the whole thing is backed by a major trusted player in the Linux world.

Rancher also provides a whole range of additional tools for container management – it's not just Docker!

	Docker Desktop	Rancher Desktop
Build, push, and pull container images	✓·	~
Run stand-alone containers	~	~
Works with external tools like Visual Studio Code	~	1
Choose the version of Kubernetes you want	×	~
Test how workloads handle Kubernetes upgrade	×	~
Choose between containerd and dockerd	×	/

Rancher Desktop can be a good alternative to the costly Docker Desktop.

74 | XF311 February 2024 www.linuxformat.com

HDDs up, this is a RAID!

Handy admin tips for locating those many drives in your redundant arrays.

ometimes novice administrators don't understand that cloud disk performance management and optimisation can be an art form. Some cloud providers limit individual disk I/O performance to ensure decent performance is available to all.

As an example, in Azure, as disk sizes increase, I/O allowances don't scale linearly. Those cloud vendors want you to buy the much more expensive SSD rather than the cheaper HDD. Disks can get very

expensive as performance and size increase.

Fear not – there is a way around this issue that keeps performance up and costs down: using the Linux mdadm command (software RAID). This enables the administrator to stripe a group of less performant disks together and get the performance of those individual disks combined without affecting availability (loss of one disk). In effect, it creates a RAID 1 disk stripe using software RAID.

As an example, rather than having one 1TB disk with 60MBit/s and 120 IOPs limitation, you can use mdadm and merge four 256GB disks with more generous allowances per GB, and quadruple the performance for just a few dollars more – and substantially less than the SSD equivalent. This can be the difference between acceptable performance on normal HDD and SSD.

The great thing about cloud disks is that the real physical disks underlying the cloud are (or should be!) already redundant at the hardware level, so there is no risk in using RAID 1 (no disk redundancy) in the cloud VM. A word of caution: use this only for data disks, not boot disks. It may be possible, but it's redundant because the first disk should only contain the essentials. As previously mentioned, the disks are already redundant at the real hardware level.

Implementing such a useful feature in Linux is quite straightforward. Using Azure as an example, create an Azure VM with a boot disk and make sure there's an additional four data disks. Once the VM is up and

mdadm: Note: this array has metadata at the start and
may not be suitable as a boot device. If you plan to
store '/boot' on this device please ensure that
your boot-loader understands md/v1.x metadata, or use
--metadata=0.90
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
sysadmin@lf1:~\$

NAME	MAJ:MIN	DΜ	STZE	PO	TYPE	MOUNTPOINTS
Loop0	7:0	0	59.2M	1	Loop	/snap/core20/1977
Loop1	7;1	0	109.6M	1	loop	/snap/lxd/24326
Loop2	7:2	0	46.4M	1	loop	/snap/snapd/19459
sr0	11:0	1	1024M	0	rom	
vda	252:0	0	64G	0	disk	
⊢vda1	252:1	0	1G	0	part	/boot/efi
−vda2	252:2	0	2G	0	part	/boot
∟vda3	252:3	0	60.9G	0	part	
└ubuntuvg-ubuntu1v	253:0	0	30.5G	0	lvm	
vdb	252:16	0	5G	0	disk	
vdc	252:32	0	5G	0	disk	
/dd	252:48	0	5G	0	disk	
vde	252:64	0	5G	0	disk	

Finding all the appropriate disks to add in the array.

running, login via SSH and check the disks are available by using sudo lsblk to make sure those four additional disks are present and not mounted already or otherwise in use.

With that list of disks, creating the new RAID 1 disk becomes quite straightforward, we'll explain.

Once you've verified which disks are the data disks, use the command below to create the virtual disk: \$ sudo mdadm --create /dev/md0 -level=0 --raid-devices=4 /dev/vdb /dev/vdc /dev/vdd /dev/vde

Using this creates /dev/md0 - a virtual RAID disk that is made up of our data disks /dev/sdb, /dev/sdc, /dev/sdd and /dev/sde. Let's ensure the array is automatically assembled during boot time with:
\$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/

Finally edit **fstab** to add the array so that it automounts on boot. Use sudo nano /etc/fstab and add: /dev/md0 /mnt/md0 ext4 defaults,nofail,discard 0 0

Once the disk is created, the administrator can use it in the same way as any other physical disk, because it shows up as /dev/md0. We suggest using LVM to manage it appropriately. Using LVM makes it trivial to set up custom disk sizing for different uses.

Finally, and very importantly, when setting up the filesystems to mount at boot, make sure the **fstab** file is valid by using **sudo mount** -a . This command rereads the **fstab** file and makes sure the entry is valid, as well as mounting the disks in the **fstab**. This step is very important because if the **fstab** file has an error in it, it may very well prevent booting into multi-user mode on reboot, which means SSH won't be up and it becomes a pain to fix.

It may or may not be worth your effort. It depends on the workload versus cost, but it can be useful for production scenarios where you need performance at the best possible cost. A last word of caution: each Azure SKU had a finite number of disks. Something to bear in mind but balance that against paying a few extra dollars to quadruple the performance.

www.techradar.com/pro/linux

Creating a RAID 1

disks to optimise performance.

array to use all four

AccuWeb Hosting

Power-packed web hosting for all levels of user, including the ever-so-picky **Mike Williams**.

IN BRIEF

No marketing tricks, no misleading headline prices, no 'unlimited' accounts that really aren't -AccuWeb Hosting has a strong range of quality products, and is very configurable fairly priced, honestly described and with quality support.

ounded in 2003, AccuWeb Hosting is an experienced US provider with a huge range of web hosting products and services. With data centres around the world, shared plans can be hosted in the US, Europe, Singapore, South Africa, India and Australia, and VPS and dedicated plans add even more options.

Most products are highly configurable. Looking for an unmanaged VPS? Control panel options include *cPanel*, *Plesk* and *DirectAdmin*, or the company can install free options like *ISPConfig*,

Ajenti or ZPanel. After a US dedicated server? AccuWeb has so many options that the website uses eight separate filters to help you find what you need.

AccuWeb Hosting plans don't claim to offer unlimited bandwidth and storage, but we don't see that as an issue. The limits are generally set very high; even the cheapest shared hosting plan includes 10GB storage and 500GB monthly bandwidth, probably enough for most consumer and many business sites. And in reality, all providers have limits: they just hide them behind vague fair usage clauses.

Headline prices aren't the lowest, but they're very reasonable (shared hosting from \$3.99 a month on the three-year plan, VPS from \$10.99 on an annual plan), and they often get better when you drill into the details. You're also protected by a 30-day money-back guarantee (seven-day for VPS).

Secure sign-up

Creating an account is a little more secure than usual (or involves more hassle, depending on your point of view) – the company validates your identity by sending you a text with a one-time password. Once verified, AccuWeb Hosting activated our account at speed, and within 15 minutes a Welcome email arrived with logins, FTP credentials, nameservers and every other detail.

Logging in took us to a familiar WHMCS-based account management dashboard, as used by the majority of smaller web hosts. This doesn't have much visual appeal, but it includes all the account, product and support features you need.

A couple of clicks takes you to AccuWeb's fully-featured *cPanel* setup, where you'll find several tools to help create a site. *Softaculous* is on hand for easy installation of *WordPress*, *MediaWiki*, *PrestaShop* and hundreds of other web apps. *RVsitebuilder*, a simple drag-and-drop website builder, is also included.

The standard *cPanel File Manager* enables creating and organising folders, uploading and editing files, and generally getting your web space in order. There are all



Shared hosting plans include cPanel as standard.

the usual website management features, too, including tools to work with domains and subdomains, create databases, create and manage email accounts, and get in-depth stats on your website visitors.

For support there are more ways to help than most: live chat, email, telephone, a web knowledgebase and some busy social media sites. There's a huge amount of content available on the knowledgebase alone.

Uptime top ranking

We used **Uptime.com** to check the availability and response time of our test site at five-minute intervals over a seven-day period (that's more than 2,000 checks). The results showed our site had 100% uptime, with no outages detected. That's what we expect for a short-term test, but some hosts don't manage it.

Response times averaged 285ms – that's the midrange of what we'd expect for basic shared hosting. (Remember, the results tell us nothing about what you might see from AccuWeb Hosting's other products.) AccuWeb Hosting scored for consistency, too. Results ranged from 248-485ms, but there are only a handful of readings above 400ms, and our site mostly returned results in a very acceptable 250-300ms range.

VERDICT

DEVELOPER: AccuWeb Hosting WEB: www.accuwebhosting.com PRICE: From \$3.99 pm

 FEATURES
 9/10
 EASE OF USE
 9/10

 PERFORMANCE
 7/10
 VALUE
 7/10

AccuWeb Hosting's limits on bandwidth and storage might put off some, but its allowances are generous, and its lengthy feature list ensures you're getting plenty for your money.

Rating 8/10

CREDIT: AccuWeb Hosting

Smartproxy

Sead Fadilpašić tests a versatile and fairly affordable proxy service.

IN BRIEF

A proxy platform that offers a wide range of features and fairly competitive pricing. With an easy-to-use dashboard perfect for beginners, it's backed up by a knowledgebase filled with helpful articles for troubleshooting and guidance. For more experienced users. Smartproxy offers an API that allows for scaling data mining tasks. However, there are keenerpriced packages out there.

martproxy is a fast-growing proxy service that has been around since 2018, and while the length of its tenure may not sound promising, the offer and capabilities are. The main service revolves around roughly 55 million monthly IPs from its residential proxy offer. There is also a 10 million IP pool derived from phones and other devices on a cellular connection. The company claims that its residential proxies are the fastest on the market, with response speeds of less than 0.6s and a 99.47% success rate. The IP addresses generally come from over

195 locations around the globe, making the IP offering as varied as possible. To round things out, its data centre proxies are IP addresses created in server hubs.

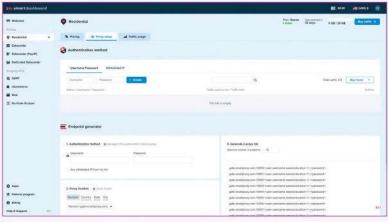
Smartproxy also offers a unique feature called Site Unblocker, which does as the name implies: lets you visit restricted sites, bypass stringent bot measures, and pull data. The firm boasts an impressive success rate of 100% for the unblocker. Users can also benefit from features such as fingerprinting, proxy rotation, pool management, JavaScript rendering and more.

The company recently introduced a new pricing scheme, with prices varying depending on your needs. The Residential Proxies plans start at \$8.50/GB for the 1GB plan and go up to a 100GB plan for \$5.20/GB. Enterprise plans have a different pricing scheme that includes a dedicated account manager with each plan, spanning from 250GB to 1,000GB. All plans (regular and enterprise) give you access to over 55 million IP addresses, an unlimited number of parallel sessions, HTTPS & SOCKS5 support, rotating and sticky sessions, 24/7 tech support, and country and city-level targeting.

For simplifying web crawling, Smartproxy offers various scraping tools. The SERP Scraping API helps gather real-time data from search engines for all SEO-related metrics, while the eCommerce Scraping API does the same for major ecommerce sites and provides a thorough analysis to help follow market trends.

The Web Scraping API enables users to collect publicly accessible data from even the most complex websites built with JavaScript. Finally, the Social Media Scraping API can scrape social media platforms of any scale and obtain well-structured data in raw HTML files or parsed JSON results. The social media tool covers all popular platforms (Instagram, TikTok, X and so on) and can offer the results in an HTML or GraphQL format for analysis. For the less experienced, the No-Code Scraper enables you to scrape data using pre-made templates.

There are also browser extensions in the toolkit. By using its *Chrome* proxy extension, you gain immediate



The newly updated Smartproxy Dashboard.

access to the intermediary. The plugin allows for custom sessions with just two simple clicks, and it is compatible with various types of devices. If you have a valid Smartproxy plan, you also receive a free *Firefox* add-on. This extension prioritises anonymity, localised content and effortless website unblocking. Its design ensures a hassle-free experience.

Another convenient feature provided by Smartproxy is the Smart Wallet. This feature simplifies the process of purchasing proxies by enabling users to store all proxy-dedicated budgets in one place. It is particularly beneficial for users who prefer making payments with Apple Pay or Google Pay.

Smartproxy provides documentation consisting of a wide range of helpful guides neatly categorised based on the platform's various features. To facilitate access to information, users can utilise the search function by using keywords. The documentation is designed with user-friendliness in mind, featuring articles written in simple language that's easy to comprehend and follow. Should users encounter any difficulties or experience issues, the support team is available 24/7 through live chat and email.

VERDICT

DEVELOPER: Smartproxy **WEB:** https://smartproxy.com

PRICE: \$8.50/GB

 FEATURES
 8/10
 EASE OF USE
 8/10

 PERFORMANCE
 8/10
 VALUE
 7/10

Smartproxy is a versatile and fairly affordable proxy service provider that offers a range of proxy options, reliable support and user-friendly features.

Rating 8/10



The world's biggest PC maker says it can achieve net zero by 2050. Barry Collins visits its US headquarters to get his hands on plant-based PCs, bamboo packaging and vegan leatherstyle covers.

t's not easy being green, as a wise young frog once sang. It's definitely not easy if you're a global PC manufacturer in an industry where the trend has emphatically swung towards sealed, largely unrepairable devices over the past decade. How far can a company such as Lenovo swing

the pendulum back? That's what we were invited to the company's US headquarters in North Carolina to find out. And before you point out the irony of flying around the globe to cover green issues, personal flight CO₂ pales in comparison to the footprint of manufacturing.

There we saw how Lenovo was making greater use of recycled materials, manufacturing laptop cases from flax, packaging from bamboo, computers that are designed to last longer, and servers cooled with water instead of energy-hungry air conditioning. They're all part of its goal to become net zero by 2050, with some stiff targets to meet in the much shorter term, too.

Can a top-tier PC maker shifting tens of millions of PCs every year really make zero contribution to global greenhouse gas emissions within 30 years? Nobody can know for sure, but there's zero doubt that the company is at least taking steps in the right direction.

Recycled PC cases

Lenovo sells an awful lot of PCs – almost 69 million of them (some of them even run Linux!) in 2022, according to research firm Gartner, making it the biggest box-shifter in the world by quite a margin. It accounts for just under a quarter of global PC





shipments, so can have a sizeable impact on resources if it can make better use of recycled materials and create less waste. Not least because, as market leader, it can set an example for others to follow.

There's upward pressure on the company to do less damage to the environment, too. "Every customer I speak to says, 'Do you have a target for driving zero carbon emissions?" says Tom Butler, the company's executive director of commercial portfolio and product management (Lenovo could make serious environmental savings on business card printing by cutting the length of its job titles).

One of the ways Lenovo is reducing waste is with the increased use of natural or recycled materials for PC and laptop cases. We're taken to Lenovo's design lab, where we're surrounded by laptop case designs, stretching back to the beige IBM-branded laptops of the '80s through to prototypes of unreleased devices that are annoyingly hidden beneath a thick black sheet.

Also in the lab are prototypes of materials that Lenovo is developing for use in its forthcoming laptop ranges. There are samples of 'vegan leather', made from cactus, that could be used to cover laptops or for cases. There's carbon fibre, reclaimed from the fuselage of Boeing plane bodies before being forged into plates that each have a unique pattern, giving them an interesting design aesthetic. And there are the laptop lids for the recently launched ThinkPad Z13 Gen 2, made from harvested flax plant fibres. They're woven together with tree sap and, again, each case has a unique texture due to those woven flax fibres.

Tom Butler wants to demonstrate that the flax covers are no less durable than their plastic or metal equivalents, and he does so emphatically, by dropping the laptop from waist height and then standing on it. "Now turn it on," shouts one of the journalists in the room, and there's an awkward moment when he presses the power button and nothing appears on screen. He hands the ThinkPad to a colleague who, having not left the room with the eyes of a dozen journalists fixed on him, brings the working laptop back to Butler a minute or two later. "Unexpected Windows update," the colleague explains.

Vegan leather and flax covers are all well and good, but the vast majority of the product range will still be made from plastic or metal for the foreseeable future. Here, too, however, the company claims to be making good use of recycled materials. Every laptop in the ThinkPad range contains some post-consumer content (PCC) materials, Butler claims. For example, the covers used for the top and bottom of some ThinkPads use 75% recycled aluminium.

Why not 100% recycled materials across the board? "As you use recycled material, it loses some of its strength and durability properties," Butler explains, potentially harming the lifespan of the laptop, which would be counterproductive. "Plastic PCC breaks down – I can use it for five generations and then it degrades."

The company tests recycled material to find out how many generations of recycling it's been through – the properties of the plastic change with every generation of reuse. However, new life can be breathed into old plastic by mixing it with fresh materials. "We have to use a blend of recycled content as well as some new content," says Butler. "That's why you don't see 100% recycled content."

Plastic-free packaging

It's not only the computers themselves that potentially create plastic waste, but the packaging, too. Although not for much longer. All ThinkPad packaging will be 100% plastic-free by the end of the year, according to Butler. "Even down to the security label that goes over the package," Butler adds.

Up in the design lab, we're shown how Styrofoam packaging designed to protect the laptop from shocks and bumps in transit has been replaced by bamboo inserts. Why bamboo? It grows exceptionally quickly, requires relatively little water to grow, and it's compostable. To prove it, Butler holds up a bamboo packaging insert that he laid on the soil in his own garden a year ago. We say holds up, but what's left of the bamboo is actually contained

of the bamboo is actually contained within a clear plastic folder, because it has largely disintegrated.

The commitment to environmentally friendly materials stretches to the ink used to print the company's logos on packaging, where it's experimenting with algae-based ink. The big problem with the algae inks at the moment?

They're struggling to find a

Lenovo is now making laptop cases from flax.

Left: ThinkPad

Right: Modular

barriers persist.

packaging will be

100% plastic-free

by the end of 2023.

laptops are popular with customers, but



February 2024 LXF311 79

IN DEPTH Green PCs





Lenovo is reducing waste by using recycled materials in its cases.

pigment that matches the trademark IBM/Lenovo red that you see on the keyboard TrackPoint, for example.

Making PCs last longer

Lenovo's laptops already have a strong reputation for durability, even in this writer's household. A ThinkPad that fell out of a rucksack and skidded across a London road remains in active use as his daughter's day-to-day laptop five years later. However, the company admits it's got more to do to make its products longer lasting and more repairable.

From the moment Steve Jobs slid the first MacBook Air out of a manilla envelope, the laptop industry has been obsessed with slender, sealed units that are often difficult, if not impossible, for professionals to repair, let alone individuals. "The primary constraint is what the market demands in terms of form factors and functionality, weight, battery life, all those sorts of things," admits Lenovo's senior technologist, Kevin Beck, when we push him on why laptops have become less repairable over the past decade or so.

But he pushes back on areas where Lenovo has resisted some of the industry's least environmentally friendly practices. "We do not use any soldered-on SSDs," he says, at least in the company's business-orientated products. "Primarily because there's a large number of regulations and company policies around data retention. If they have personally identifiable customer or patient data on it, they can't allow that data, even encrypted, to leave their premises. So, when they have to send something off to be repaired or replaced, they want to be able to take that [SSD] out."

Likewise, Beck insists components such as screens aren't glued down. "There has been adhesive tape," he concedes, "but it was removable and included in the repair kit for the technicians," though you couldn't

necessarily buy that tape on the open market.

What about soldering down RAM? "Soldering of RAM is driven primarily by form-factor concerns," Beck responds. "We have a mix. Some of it is onboard and part of the motherboard, but part of that is that the connectors are industry standard, and they are of a certain thickness. We can't go off, and nor do we want to go off, and design some

brand new, proprietary memory interface. So that is a constraint, but I know we are looking at ways of making it better."

Beck insists repairability remains a key priority for the company. Indeed, by 2025, Lenovo set a very specific target that 84% of PC repairs should be eligible for on-site customer service rather than having to be shipped back to a depot, with all the inconvenience, cost and transport emissions that creates.

To that end, the company makes the hardware maintenance manual of every ThinkPad available online – it has done since day one – and also produces YouTube videos to help customers repair or upgrade its machines themselves. ThinkPads only use standard screws and tools, so there's no artificial barrier to repairs. He gives an example of how the company evolves products to ensure they're more repairable. The first foldable ThinkPad, the X1 Fold, "did not pass our own criteria for onsite repair. It was repairable to an extent, but it had to be sent back to us." For the second generation, the company changed the cover design and made it detachable by removing a screw and then popping a thin screwdriver into a hole on the side.

The company has also been doing work on the beep codes that indicate a serious hardware fault. Instead of relying on customers to work out whether it was, for example, one long and two short beeps or three short beeps when diagnosing faults, the company created a smartphone app, so customers could place their phone by a faulty PC and have it diagnose the problem by listening for the beeps. However, the company realised that asking already harassed customers to download a separate app was making a bad situation worse, so now customers are asked to hold their phone near the faulty laptop when talking to a service centre, with the phone system able to do the same job the app did.

Developing these things comes at a cost, but as Beck explains, "When we look at how much it costs to make a product, we look at the lifespan of the product, including the warranty. There are places where we've decided to spend a few cents up front ... to make it cost-neutral, to reduce the amount we spend to repair it over the course of its life."

Making modular laptops

Up in the labs, one of the ideas Lenovo is kicking around is making laptops more modular. Granted, this is hardly radical thinking. Framework has garnered a lot of attention and praise for its modular laptops, which provide toolless access to the internals, letting you easily replace or upgrade components such as the battery, storage or keyboard.

Lenovo knows the concept is attractive to consumers. It shares the results of a survey with us, which polled 2,000 consumers in various geographies. Just under three quarters felt repairability was very important, 59% wanted devices to be repairable by the manufacturer, and 22% wanted to be able to perform repairs themselves.

Kevin Beck says Lenovo's not about to take the exact same approach as Framework, but it is interested



in the idea. "The concept of modularity is something we're actively looking at and exploring," he says, although he quickly caveats that with some of the difficulties of shipping laptops with replaceable parts. He says that if you mapped all the interconnections between different components and assemblies "it would look like a galaxy between all the considerations of the parts, the lifecycle, the stocking costs, moving it around the world, the minimum order quantities for certain things. It just has to be balanced."

And then there's the fact that Lenovo can't do this

And then there's the fact that Lenovo can't do this alone – it needs component manufacturers to get behind the idea of modularity to really make it a success. "It is definitely something that the entire industry is going to have to row in the same direction on, to some extent," he adds.

Beck senses our doubt that Lenovo's ever going to get round to delivering a modular laptop with so many barriers being put in its path, but he ends on a more confident note. "It's hard to see the end of the road because it's so complex to know where we'll end up," he says. "I'm not equivocating, for that reason. It could just go so many ways, but the upside of that is there's a lot of talk, there's a lot of good possibilities."

Recycling components

A lot of the focus so far has been on creating new product. But finding better ways to deal with old equipment is also high on Lenovo's priority list, not least because its environmental targets stretch up to the supply chain and down to what customers are doing with Lenovo equipment.

Lenovo provides several examples of how materials reclaimed from old equipment are being reused, including in high-end jewellery. But there are only so many people who want a ring made out of old motherboards, so Lenovo is putting serious effort into reusing components that haven't yet reached the end of their useful lives. "A specific, stated part of the strategy that we're adopting around circular economy is to find ways to reuse parts from old systems in the right way, or to extract value from them in some other way," says Beck.

Beck insists old components won't be reused in new PCs, but components or whole systems can be harvested and resold as used equipment. Around 60% of the PCs brought in through the company's corporate asset recovery program are reused, an executive claims in a subsequent presentation. However, Beck says the company is also making efforts to reuse perfectly good returned parts.

"There's a surprising amount of what we call NDFs – no defect found," says Beck. "The customer thought it went wrong, it didn't work for some reason, but when they get it back in, the brand-new part [was fine]. So that's what we mean by parts reutilisation in the active servicing programme, and that is something that we have metrics on for all of our business partners and authorised servicers. It's the degree to which they're doing the right thing with the parts."

Meeting the challenge

We're in an industry with too much greenwashing. Too many spurious claims of environmental benefits, too many companies where the green commitments

>> SERVERS THAT HEAT A SWIMMING POOL

There are few things more power-hungry than data centres, and their power consumption is set to quadruple by 2030, according to Lenovo.

More than 30% of a regular server's energy is devoted to the cooling fans, which is why Lenovo is continuing to develop its Neptune water-cooling tech, a far more energy-efficient way to keep servers from cooking.

There are two types of watercooled server in Lenovo's lineup. There are fixed-loop systems, where the same water is cycled around the server and passed through a radiator, cooling the server much like a liquid-cooled PC.

Then there are open systems, where water is pumped around the entire rack, with cold water entering the system and warm water coming out. It's what happens to that expelled warm water that helps to keep the

system more energy efficient. In some instances, it's used to heat facilities or provide warm water for taps and showers. One company is experimenting with using the hot water from its server stack to heat a swimming pool, according to Patrick Moakley, director of marketing for HPC and AI at Lenovo.

Water-cooled servers do come with some drawbacks. They are typically around 10% more expensive than fan-cooled systems, according to Moakley, although he claims companies will recoup that investment with savings on their energy bills within 12-18 months. The water cooling is often more effective than fans, too, so processors can be clocked to higher speeds. Both Intel and AMD now have water-cooled-only SKUs, according to Moakley, because fans aren't capable of keeping the chips cool enough.



It takes loads of liquid to keep this little lot cool.

extend little further than the press release. Part of the reason we accepted Lenovo's invite to its US HQ was to see if that was the case here.

It clearly wasn't. Of course, we were only shown what the company wanted us to see, and we've no doubt Lenovo's commitment to profitability trumps its commitment to the environment. It's a successful business, not a charity. But we saw strong evidence of a company trying to do the right thing, with executives who seemed genuinely enthused at the idea of making the business less resource-hungry, less wasteful. Not least because it's often as good for the bottom line as it is for the environment.

It's doubtful whether any of the executives we spoke to will be there in 30 years' time to be held to account for hitting those net-zero targets. Most of them had as many grey hairs as we do. But if they manage to imbue the same enthusiasm and dedication into the next generation of managers, there's at least a fighting chance those targets will be met.

tom's HARDWARE

GET ALL THE ESSENTIAL BREAKING NEWS FOR THE TECH ENTHUSIAST!

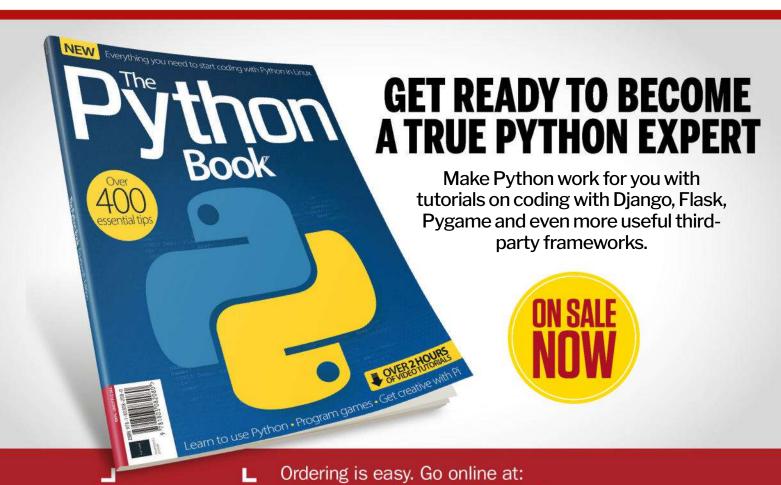
No matter if you're building a PC, buying a laptop or learning about robots, Tom's Hardware has all the comprehensive knowledge you need.



Scan & Subscribe for free!

FUTURE





magazinesdirect.com

Or get it from selected supermarkets & newsagents

THE BEST NEW OPEN SOURCE SOFTWARE ON THE PLANET

HotPicks

Recoll >> Polybar >> Curtail >> Betterbird >> BleachBit >> FileZilla >> JDownloader 2 >> Gods Deluxe >> Naval Battle >> Fantascene >> Audacity



Mayank Sharma

can't go out and risk being choked to death by the toxic air. Instead he makes do by finding apps that are a breath of fresh air.

DESKTOP SEARCH

Recoll

Version: 1.36.1 Web: www. lesbonscomptes.com/recoll

he find command is pretty handy for finding files embedded in your filesystem. Even if you don't want to get down to the CLI, most modern file managers have a find function that's dexterous enough to get the job done. But neither of these offer the functionality you get with Recoll, which can dig up documents based on their content, even inside email attachments and compressed archives.

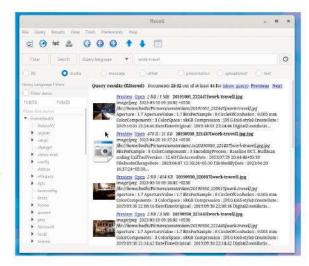
Recoll only offers source packages, and installing them takes some doing. Luckily, Recoll is available in the official repositories of most desktop distros. Fedora users, for instance, can install it with sudo dnf install recoll.

There's also a PPA for Ubuntu users, which also works on Mint. Add the repo with sudo add-aptrepository ppa:recoll-backports/recoll-1.15-on, then use sudo apt-get update to refresh your repos, before you install Recoll with sudo apt install recoll.

When launched for the first time, Recoll asks you to create an index. You can configure the index creation process by adjusting the directories you want included or excluded from the process. However, the app itself recommends to go with its sensible default settings, at least for the initial build.

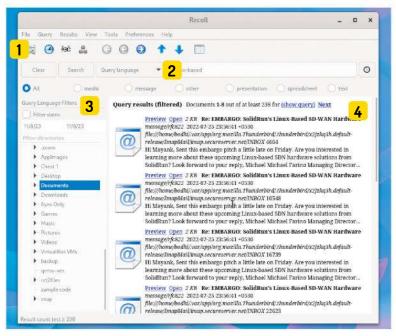
Depending on the size of the filesystem, creating the index could take a while. Recoll can index many document types, such as plain text, HTML, OpenDocument, emails and a few others on its own. For other types, you need to install external helper apps. After building the index, Recoll displays a list of missing helpers, which can also be found by going to Tools > Missing Helpers.

To find a file, enter the search term in the text box at the top, and hit Search. Recoll has several search modes. Query language should work for most cases, but you can also use the app to exclusively look for filenames, not contents. You can also use wildcards in any of the search modes. Peruse through the app's extensive documentation to make the most of it.



Recoll has plugins for Gnome and KDE that you can use to integrate it with the launchers of both desktop environments.

LET'S EXPLORE RECOLL..



Complex searches

Clicking this launches a new GUI that can help you further refine your search parameters by specifying multiple clauses and filtering parameters.

Search mode

Query language is the default search mode, which looks for documents that contain all of the search terms. Use the drop-down to select a different mode.

You can choose to filter the results by document type, or use the Filters panel to filter the results by dates or by filesystem location.

View results

You can preview or open the files that have been found with your search simply by clicking on them from inside the Recoll window.

February 2024 LXF311 | 83 www.techradar.com/pro/linux

STATUS BAR

Polybar

Version: 3.7.0

Web: https://polybar.github.io

ost of us don't really think about the status bars that ship with our desktop environment. The two kinds of users who do are either people who are really upset about them, or those who custom-build their desktop environment from scratch on top of a lightweight window manager. If you're either of those types of user, you can use Polybar to create your very own custom status bar.

You can build *Polybar* from source, but the app itself recommends installing it via the official repos of your distro. Ubuntu users can use sudo apt install polybar, while Fedora users can install the app with sudo dnf install polybar.

Polybar can display the most commonly used services for your convenience, including workspace switcher, playback controls, network connection details, CPU and memory load indicators, battery display, date and time, and more.

Polybar is a CLI utility, in that you have to configure it via text files. However, that shouldn't be a problem for anyone willing to put in the effort to replace their



Before you bring up Polybar's custom status bar, hide your desktop's existing one, using the appropriate settings or shell extension.

existing status bar or even create their own desktop from scratch.

The app places a default configuration file under /etc/polybar/config.ini. You can use this config file as is by running the polybar command.

To get started with your own customisation, you can either start from scratch or copy the existing configuration file to ~/.config/polybar/config.ini. You can then edit it as per your needs. It's fairly descriptive, but the project also has detailed documentation on its website to guide you.

For starters, you need to add a section in the config file for each bar you want to define, as you can define multiple bars in the same config file. Just make sure, you specify the name of the bar when launching the app, such as polybar <name-of-the-bar>.

IMAGE COMPRESSOR

Curtail

Version: 1.8.0

Web: https://apps.gnome.org/Curtail/

ompressing images doesn't really make sense, until you start running out of disk or cloud space, or you have to upload images in web forms that have strict size restrictions. Whatever your need, *Curtail* does a very nice job of squishing images, and supports those in JPEG, PNG, WEBP, and SVG formats.

Curtail is essentially a Gnome app, but it's available as a Flatpak, which means you can use it on any desktop without any trouble. If your distribution can install Flatpaks, fire up a terminal and use:

\$ flatpak install flathub com.github.huluti.Curtail to install the app.

The app has a straightforward interface. Just drop in the image you want to compress, or use the folder icon in the top-left corner to browse to an image or images. The app also has a Bulk Compress option lurking under its hamburger menu that you can use to recursively compress all images inside a particular directory.

By default, the app is set to create lossless compressions, which doesn't really do much



After compressing the images, Curtail shows you the size difference between the original and the compressed images, in both actual size and percentage.

compression. Slide the toggle in the main screen to Lossy to see meaningful compressions. The app runs in safe mode out of the box, which means it saves the new compressed version as a new file. By default, it uses the **-min** suffix on the compressed images.

You can change the suffix or even ask the app to replace the original file with the compressed one in the app's preferences. From here you can also override *Curtail's* default ability to preserve the original's file attributes, and even prevent it from transferring the metadata from the original file to the compressed image. If you know what you're doing, you can switch to the Compressed tab in the preferences to modify the compression levels for the different supported image types.

84 LXF311 February 2024 www.linuxformat.com

EMAIL CLIENT

Betterbird

Version: 115.4.2-bb17
Web: www.betterbird.eu

hat Firefox is to web browsers, Thunderbird is to email clients. Or at least, that's what Mozilla would like. While it is quite popular, and a very capable email client, Thunderbird is often criticised for being slow to add new features, or even fix bugs.

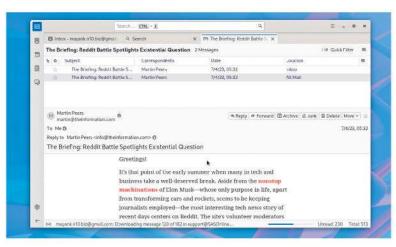
There are several *Thunderbird* forks, and *Betterbird* is one of them. Unlike some of the others, *Betterbird* is a soft fork as it closely follows the development of *Thunderbird Extended Support Releases (ESR)*.

As you can probably tell from its name, *Betterbird* bills itself as a fine-tuned version of *Thunderbird* that offers new features and bug fixes, which, in the true spirit of open source, it ships upstream as well.

Grab the *Betterbird* binary archive from its website and extract it with tar xjfv betterbird-115.4.2-bb17.

en-US.linux-x86_64.tar.bz2. Then switch to the extracted directory and fire up the *Betterbird* client with ./betterbird.

On first launch, you're asked for the login credentials of your existing email client, which the app promises to house on your computer. If you've used *Thunderbird* (or



any email client for that matter), you'll feel right at home with *Betterbird*. Apart from a few minor tweaks, even *Betterbird*'s user interface is very similar to that of its progenitor.

One of the biggest features in the new release is the multiline view, which *Betterbird* has been refining since 2021. This layout isn't enabled by default, since it isn't everyone's cup of tea. The app also has an impressive search function that supports regular expressions and can be used to build complex search queries.

In addition to adding new features, *Betterbird* is also restoring some of the features dropped upstream, such as the count of total and unread messages in the app's status bar, and the ability to select multiple folders for bulk processing.

Besides adding new features, Betterbird is fully compatible with the gazillions of add-ons available for Thunderbird.

SYSTEM CLEANER

BleachBit

Version: 4.6.0

Web: www.bleachbit.org

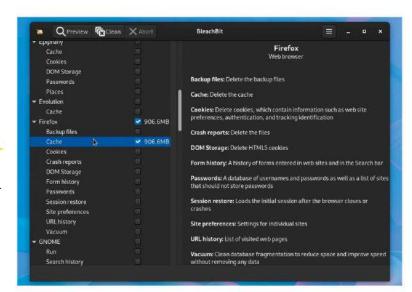
ur distros accumulate a lot of digital gunk over time, but *BleachBit* helps you spring clean your computer as well as protect your privacy. It can remove temporary and other unnecessary files, and also has tools to securely delete files or wipe free space to ensure they cannot be recovered.

The version of *BleachBit* in the repos of most distros is often stale, and the project suggests you use the precompiled binary for your distro from its website.

On first launch, *BleachBit* asks you to review the app's preferences. It's a good idea to do so, but it ships with reasonable defaults, so you can safely jump straight in as well.

The app is designed to free up space on your computer by removing unwanted content. *BleachBit* can empty the cache, and remove cookies and cache files from apps such as *Firefox* and *LibreOffice*. It can also empty the trash, delete temporary files, wipe the swap memory, delete old system logs and a lot more.

The *BleachBit* GUI is divided into two frames. On the left you have a list of the installed apps, which expands



to reveal the list of all kinds of data it houses. Click on any to get a brief explanation of each one of these togglable options in the right frame.

To clean an area, such as *Firefox*'s cache, simply click on the checkbox next to it. Completing most tasks doesn't take much time, but some operations, such as the option to overwrite free disk space to hide deleted files, do take a while, and the app warns you whenever you select such a task.

For all its benefits, exercise extreme caution when using *BleachBit*. If used impulsively, the app can remove necessary files and destabilise your installation.

Before you ask BleachBit to zap the useless files, it's always prudent to use the Preview button to review the list of files it'll delete. **FTP CLIENT**

FileZilla

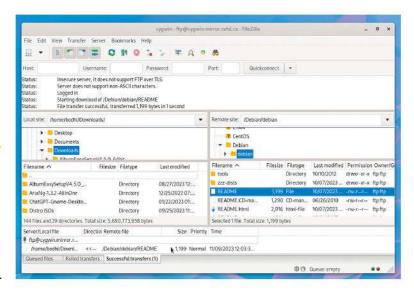
Version: 3.66.1

Web: https://filezilla-project.org

here was a time when an FTP client was the best way to grab Linux distros, and many also shipped with one as well. Sure, it's still one of the best ways to host files, but the chances of everyday desktop users running into an FTP link is fairly slim. But if you're one of the handful who grabs stuff via FTP on a regular basis, you'll want to use FileZilla.

The project puts out a build that's made for Debian 10. For users of all other distros, the project recommends either compiling it manually or using the version that ships with your distro. Compiling from source isn't for the faint of heart, so it's best to just grab the app with sudo apt install filezilla if you're using Ubuntu, or with sudo dnf install filezilla if you use Fedora.

File Zilla is essentially divided into two panes. On the top of the left pane you can view your filesystem, with the files and directories in the selected location shown in the bottom. Similarly, the right pane shows the filesystem and the contents of the currently selected directory of the remote FTP server. For a one-off connection, you can enter the servername and its login



credentials in the app's main window. If you need to connect to this server regularly, head to File > Site Manager. Here click the New Site button, and key in the details of the server.

This screen also lets you choose the protocol you want to use for the connection. Besides plain FTP, which is insecure, *FileZilla* also supports the other commonly used, secure, FTP protocols, namely SFTP (SSH FTP) and FTPS (FTP over SSL/TLS).

Once connected, right-click on the file and select the Download option to transfer it to the currently selected directory in your machine.

Not all servers support the secure FTP protocols, and FileZilla warns you before it sends your passwords in cleartext.

DOWNLOAD MANAGER

JDownloader 2

Version: 48254

Web: https://jdownloader.org

alking of downloading stuff, your Linux distro already offers plenty of tools. There's Wget for anyone not afraid of the CLI, while those looking for a graphical option can get decent mileage from the downloader in their web browser. While these options are good for the occasional download, if you are always grabbing stuff from the internet, you need to do it efficiently with a download manager, and JDownloader 2 is one such downloader.

There are several ways you can download JDownloader, but the most convenient is using the Flatpak. Download it from the terminal with flatpak install flathub org.jdownloader.JDownloader . Don't worry about the version, because the app by design queries its server and automatically grabs the latest release. If you have JDownloader running, the app lets you know when a new update is available with a flashing update button in the toolbar.

The app is written in Java, which means it doesn't look very pleasant on modern desktop environments. But if you are willing to look past its appearance, you'll



In addition to regular files, JDownloader can also download torrents, individual videos from YouTube, or even entire channels.

find the app isn't lacking in features. Sure, its interface is a lot busier than your average download manager (and it's got in-app ads), but it's very intuitive and should be fairly easy to operate for anyone who has worked with a torrent client.

Once you have *JDownloader* running on your system, all you need to do is copy a download link from your browser. This transfers the link to the distro's clipboard, from where *JDownloader* automatically picks it up and adds it to its download queue. You can then switch to the app and press the Play button to begin the download.

Besides the clipboard, *JDownloader* can even parse text files or web pages to hunt for downloadable links.

86 LXF311 February 2024 www.linuxformat.com

JUMP 'N' RUN

Gods Deluxe

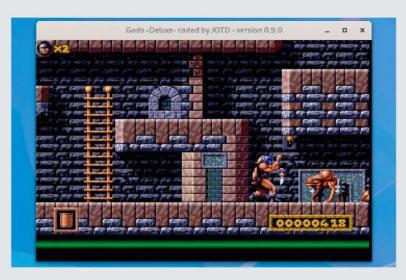
Version: 0.9.0 **Web:** https://github.com/jotd666/gods-deluxe

ods Deluxe is a remake of the '90s classic platform game for the Amiga and Atari ST. The author of the open source remake has recreated all the levels of the original, and then gone a step ahead and added four new levels of his own.

To play the game, you first need the OpenJDK 1.7 runtime environment. It's available in the official repos of most distros. Ubuntu users can grab it with sudo apt install openjdk-17-jd, while Fedora users can use sudo dnf install java-17-openjdk.

After installing OpenJDK, grab the game's compressed archive from GitHub, and extract it with unzip gods-deluxe-0.9.0.zip. Then switch to the unzipped folder and run the game with ./gods.sh. The game runs in windowed mode by default but you can use the -full-screen option to run the game full-screen.

Gods Deluxe begins with an intro video that explains the player's objective. You must venture into the ancient city to reclaim the citadel that the four guardians have stolen from the gods. In addition to porting the game to the PC, the developer has



reworked the graphics while trying their best to be faithful to the original game.

Before jumping in, head to the options screen to tweak it as per your liking. There are four game modes or sets, and three difficulty levels. You can also choose from two different tilesets and just as many liveries for the protagonist. Instead of the Enter key, the game uses Ctrl to select an option.

Once inside the game, you can use the Ctrl key to throw weapons at the enemies. You can also jump on them and turn them to pulp. On the way, you collect power-ups and items such as keys in order to progress through the levels.

In addition to English, the game's menus and the in-game text have also been localised to French, German, and Hungarian.

BOARD GAME

Naval Battle

Version: 2.1.23082 **Web:** https://apps.kde.org/knavalbattle/

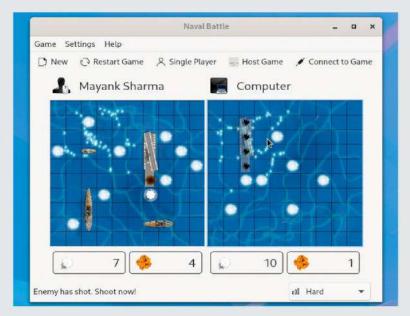
aval Battle is a two-player strategy board game where the objective is to sink the opponent's four battleships. Although it's a KDE game, you can install it on any desktop environment if your distro has the ability to install Flatpaks. If it does, use flatpak install flathub org.kde. knavalbattle to install the game.

The game display two grids when you start. The left one belongs to you, and the right to your opponent. Check the game's status bar for instructions. At the start it asks you to place your ships on your side.

The blue area in the grid represents water. You can place the ships by hovering and clicking over your grid. By default, the ships are orientated horizontally, but you can right-click to orientate them vertically.

The ships come in four sizes. The smallest occupies a single quadrant, and the largest takes up four. Note that you can't move the ships once you've placed them.

You can't see your opponent's ships, nor can they see yours. Once the ships have been placed, target your opponent's ships by clicking on a quadrant in their



grid. Players take turns to fire. A ship is sunk when all the quadrants it occupies are hit. Once you hit a ship, shoot in the adjacent quadrants as they are likely to also be occupied by the ship. The game lets you know when all the quadrants a ship occupies are hit. The player who sinks all their opponent's ships wins.

You can play the game against the computer or another player over the network. Use Host Game to start a network game. Your opponent must use Connect To Game to find and connect to your game. Place your ships carefully to make them harder to hit. Also, think carefully before taking a shot to maximise chances of a hit.

www.techradar.com/pro/linux February 2024 LXF311 | 87

WALLPAPER CHANGER

Fantascene

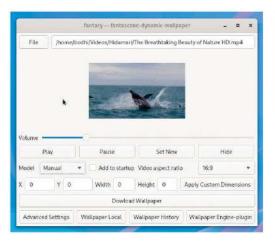
Version: 1.6.1 **Web:** https://github.com/dependon/fantascene-dynamic-wallpaper

e won't hold our breath for mainstream distros to offer video wallpapers. It's just a novelty but, boy, do these look good! So, if you run Linux on a modern machine with a multicore processor, oodles of RAM, and dedicated graphics hardware, take *Fantascene* for a spin.

The project releases through Open Build Service (OBS) and offers binaries for various releases of Debian, Fedora, OpenSUSE and Ubuntu. You can either download the binaries for your distro directly or by adding their repo. The project's OBS page linked on its website offers step-by-step instructions for adding the relevant repo and installing the app for every supported release of the four distros mentioned above.

The app has a simple user interface. You simply point it to a video and hit Play. The app then replaces your current wallpaper with the video. If the video has an audio track, you can also adjust its volume from Fantascene's interface.

Fantascene ships with a sample video of dolphins to give you a feel for the app and experiment with its



Fantascene can start when you boot into the desktop, and the app has a list of websites from where you can grab video wallpapers.

settings. Use the File button on top to point the video to any MP4 file. Whenever you point the app to a new wallpaper, press Set New to activate it. There's also a self-explanatory Pause button.

The Hide button hides the app's interface, however, instead of taking down the video wallpaper as you might assume. In fact, there's no apparent option to restore the old static wallpaper. After some fiddling around, selecting Manual from the Model pull-down menu took down the video wallpaper, but we aren't sure if that's the correct way to go about it.

The app displays desktop icons by default but you can override that behaviour from its advanced settings, from where you can also tweak the video FPS, and more.

AUDIO EDITOR

Audacity

Version: 3.4.0

Web: www.audacityteam.org

udacity is one of the crown jewels of open source software. If you need to work with audio, whether you want to put together a podcast or just record and edit audio, there's really no better option.

The app ships as an Applmage, which you can grab from its website. Then just make it an executable, either from the file manager or with chmod+x audacity-linux-3.4.0-x64.AppImage, and double-click on it to launch the app.

The app scans for plugins on launch. In our case, it found a couple of incompatible plugins, which it promptly disabled to prevent *Audacity* from misbehaving, although we did have the option to enable them if we wanted.

The interface might seem intimidating to a first-timer, but then again, so would any specialised app. To get started, either drag an audio file you want to edit into the main window, or smash the Record button to start recording from your computer's built-in microphone. You can then use *Audacity* to trim audio,



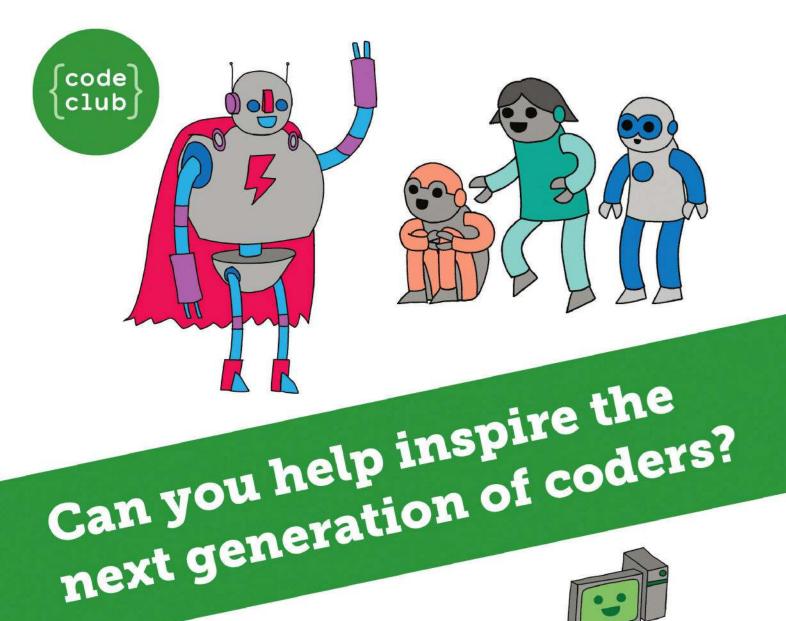
combine tracks, and even stack multiple tracks, as well as export to a number of formats and quality settings. In addition to its built-in features, the app has a whole load of plugins and add-ons that bring in even more functionality.

Being a point release, the latest version of *Audacity* brings in several interesting new features. One of the two that got our attention is the new beats and measures grid that you can use to better visualise the tempo and rhythm, and align tracks accordingly. The other is the use of a new time stretching algorithm that allows changing the duration of clips without harming the pitch.

The project's documentation has also been updated and we suggest you refer to its manual when using the app. **LET**

Audacity supports every popular audio format and even lets you combine audio from different formats into the same project.

88 LXF311 February 2024 www.linuxformat.com



Code Club is a nationwide network of volunteer-led after school clubs for children aged 9-11.

We're always looking for people with coding skills to volunteer to run a club at their local primary school, library or community centre for an hour a week.

You can team up with colleagues, a teacher will be there to support you and we provide all the materials you'll need to help get children excited about digital making.

There are loads of ways to get involved!
So to find out more, join us at www.codeclub.org.uk

CODING ACADEMY



Write your own Android application

Matt Holder investigates how to create cross-platform applications using a combination of Python and the Kivy framework.



Matt Holder

is an IT professional of 15 years, Linux user for over 20 years, user of plenty of home automation gear and selfprofessed geek. n this article, we are exploring *Kivy*, learning about its GUI language and how this links back to Python code and developing an application that can run on multiple platforms. *Kivy* is supported on Windows, Mac OS, Linux, Android and iOS.

By the end of this article, we will have developed an app that runs on desktop Linux as well as Android. We will also have introduced the necessary concepts to create a simple stopwatch application, rendered our simple

user interface, written the logic to update the text and perform the counting function, as well as compiling an APK that can be installed on Android devices.

The complete code can be downloaded from https://github.com/mattmole/LXF310-Kivy-Stopwatch. So, without further ado, let's begin.

The first thing to introduce is *Kivy's* design language and how this links back to the Python code. As in other GUI frameworks, GUI elements can be defined within a supporting file, which means that the Python code itself can be less cluttered, due to the GUI elements being defined elsewhere. *Kivy* does seem to have more of a crossover with the code as variables from the Python code can be referenced within the KV design file. For example, you could reference a variable in the Python code, which is used to calculate the size of the window, based on the screen resolution and the platform that the code is running on. Conversely, it is also possible to link Python variables to items in the design file, so they can be updated accordingly.

Within Kivy, we can think of our GUI as a hierarchical structure. For example, a window with a single label can be thought of as the label being the child widget of the window. Expanding this a little, thinking of our stopwatch GUI, the window contains a vertical **BoxLayout**, which is a child of the window, and the first



Our stopwatch is running on an Android phone, using the excellent Buildozer tool - see the Android Support boxout (page 93) for further information.

child the **BoxLayout** has is a label. The second child of the **BoxLayout** is another **BoxLayout**, this time in horizontal mode, which has two children, which are both buttons.

The built-in behaviour of *Kivy* is that the window automatically adjusts to a change of window size. This is accomplished using size and position hints, which are relative measurements, based on sizes of other widgets on the GUI.

Kivy uses built-in layouts to allow any GUI to be created, and these are AnchorLayout, BoxLayout, FloatLayout, RelativeLayout, GridLayout, PageLayout, ScatterLayout and StackLayout. Some of these allow unconstrained positioning of widgets, while others require widgets to be relative to the position of others. You can learn more at https://kivy.org/doc/stable/gettingstarted/layouts.html.

Design file

This code sample is used to define the GUI of the stopwatch and defines the location of the items on the screen. Here we are defining the window itself, a label to show the timing, and two buttons to reset, start and pause the timer. This file is called **timer.kv** and it links directly to the name of the application class in the code. The class is called **TimerApp** and the design file

QUICK TIP

Packages for Android can be created using the following guide: https:// kivy.org/doc/ stable/guide/ packagingandroid.html #packaging -android is named in lower case, after the application class, with the word **App** removed.

#:kivy 1.0.9 <TimerWindow>: timerLabel: timerLabel stopButton: stopButton BoxLayout: spacing: 50 padding: 50 size: root.width, root.height orientation: 'vertical' Label: id: timerLabel font_size: sp(0.25*self.width/pt(1)) size: self.texture_size BoxLayout: spacing: 50 padding: 50 orientation: 'horizontal' Button: id: stopButton on_press: root.stopTimer() disabled: True size hint x: 0.5 size hint v: 0.5 pos_hint: {"center_x": 0.5, "center_y": 0.5} source: 'images/Stop.png' y: self.parent.y x: self.parent.x size: self.parent.width, self.parent.height allow_stretch: True Button: on_press: root.startTimer() size_hint_x: 0.5 size_hint_y: 0.5 pos_hint: {"center_x": 0.5, "center_y": 0.5} source: 'images/PlayPause.png' y: self.parent.y x: self.parent.x size: self.parent.width, self.parent.height allow_stretch: True

On the first line of this code sample, we define that the program is designed for *Kivy* version 1.0.9. On the next line, we define the main window, which is called TimerWindow. This is important because it links directly with the name of one of the classes in the Python code. Similarly to Python, indentation is used to define when an object is the child of another object. For example, the BoxLayout is a child of the main window, and the label is a child of the BoxLayout.

On the next two lines, we define the variable names that will link the Python code to the GUI object – for simplicity, we have kept the variable and widget names the same. The next thing we need to do is to define a <code>BoxLayout</code>, which has spacing and padding of 50 pixels and is the same size as the parent window (note the usage of <code>root.width</code> and <code>root.height</code>). When defining the <code>BoxLayout</code>, it can have an orientation of <code>Vertical</code>, where widgets are rendered above and below each other, or <code>Horizontal</code>, where widgets are rendered alongside each other.

Once the **BoxLayout** has been defined, we now define which widgets we want to be shown. The first



of these is the label, which is used to show the stopwatch as it counts time. The <code>id</code> of the label allows it to be referred to elsewhere and <code>font_size</code> defines the size of the text. By using this calculation, the font size changes based on the width of the window. The division by <code>pt(1)</code> was required to ensure the text was the correct size on Android. As the window size changes, so does the size of the text. Finally, when defining the label, the size entry sets the size including any padding as defined by the parent layout. Note how we are not defining any default text for the label in the design file.

The next widget to display in the **BoxLayout** is another **BoxLayout**, but this time with the type of **Horizontal**. The first **BoxLayout** displays the label on the first row and then the second row contains a second **BoxLayout** with two buttons, side by side. Both buttons are defined in the same way, so only the first is discussed here.

The stop button needs to be referred to from the main code, hence why it has an **id** value and is referenced near the top of the design file. Should text need to be displayed, the **font_size** argument can be

The stopwatch application is running on Pop!_OS, with a rather fetching blue theme.

QUICK TIP

You can learn more about Kivy by following this tutorial, where you can build your own Pong game: https:// kivy.org/ doc/stable/ tutorials/ pong.html

» iOS SUPPORT

Kivy provides support for compiling applications for iOS. This process is relatively involved and requires access to a Mac OS device, the brew packaging system and Xcode. While it is technically possible to run Mac OS as a virtual machine on multiple different platforms, it would be against Apple's EULA to run it on non-Apple hardware and could also lead to issues with legal liability. The required steps are briefly described below.

Using *Homebrew* and the pip command, some dependencies need to be installed. Open a terminal and enter the following:

- \$ brew install autoconf automake libtool pkg-config
- \$ brew link libtool
- \$ pip install Cython==0.29.33

The next step is to create a new *Xcode* project, which can be opened in *Xcode*:

toolchain create <title> <app_directory> open <title>-ios/title.xcodeproj

Once opened, you can use the Play button within *Xcode* to start your application within an emulator. If you want to include other libraries, this can be carried out, as such:

toolchain build numpy toolchain update <title>-ios



www.techradar.com/pro/linux February 2024 LXF311 | 91

CODING ACADEMY Android app

QUICK TIP

Why not add extra functionality, such as countdown timers with notifications and the ability to add as many different timers as you like? added, with a similar calculation to the one above. The on_press argument is now added, with the name of the Python function that is called when the button is pressed. This stop button effectively resets the counter, so can only be used once the timer has been started and then paused – this is why the button is initially disabled. The two size_hint arguments have been added next and these are used to define the size of the buttons, relative to their parent. This means that the buttons should occupy half of the space within their section of the BoxLayout. The pos_hint arguments are similar and refer to where the widget should be drawn. In our code sample, the buttons are in the middle of their section within the BoxLayout. Finally, we add the Image widget as a child of the button, which is used to draw the Stop or Play/Pause buttons within the buttons.

The **size_hint** and **pos_hint** arguments seem quite cryptic at first usage, but are incredibly important to allow the automatic scaling of the GUI to work as expected. Equivalent arguments exist if absolute referencing is preferred to relative.

The code

Before writing any code, we need to install some libraries so that we can create our application and get it to run. To install these requirements, open your terminal and enter the following:

\$ python -m pip install --upgrade pip setuptools virtualenv

\$ mkdir folder_name

\$ cd folder_name

\$ python -m virtualenv .

\$ source bin/activate

\$ python -m pip install "kivy[base]"

\$ pip install plyer

This code sample is split into discrete sections, covering things such as importing libraries and explaining each class. The code needs to be added to a file called **main.py**.

from kivy.app import App

from kivy.uix.widget import Widget

from kivy.config import Config

from kivy.clock import Clock

from kivy.core.window import Window

from datetime import timedelta

from plyer import vibrator

The main libraries required, unsurprisingly, are widgets from the Kivy toolkit. We use import timedelta

from the datetime library, along with the **vibrator** method from plyer – which allows access to the vibration functionality on our smart phones.

class TimerWindow(Widget):

def __init__(self):
 super().__init__()

self.zeroCountString = "0:00:00.000" self.timerLabel.text = self.zeroCountString

Create a timer to update the stopwatch and display

timer = Clock.schedule_interval(self.
increaseTimer, 0.001)

 $guiUpdateTimer = Clock.schedule_interval (self. updateWindow, 0.1)$

self.startCount = False

self.msCount = 0

def increaseTimer(self,dt):

if self.startCount:

self.msCount += dt*1000

def startTimer(self):

trv:

vibrator.vibrate(0.1) # vibrate for 0.1 seconds except NotImplementedError:

print("Cannot call the vibrate function as it is not supported on this platform")

if self.startCount:

self.startCount = False

self.stopButton.disabled = False

else:

self.startCount = True

self.stopButton.disabled = True

def stopTimer(self):

trv:

vibrator.vibrate(0.1) # vibrate for 0.1 seconds

except NotImplementedError:

print("Cannot call the vibrate function as it is not supported on this platform")

if not self.startCount:

self.msCount = 0

self.timerLabel.text = self.zeroCountString

def updateWindow(self,dt):

if self.startCount:

The files and folders used within the project can be seen here.

```
LXFStopwatch-0.1-arm64-v8a_armeabi-v7a_x86_x86_64-debug.apk
buildozer.spec
images
PlayPause.png
Stop.png
main.py
pyvenv.cfg
requirements.txt
timer.kv
```

92 LXF311 February 2024 www.linuxformat.com

Android app CODING ACADEMY

delta = timedelta(milliseconds=self.msCount) self.timerLabel.text = str(delta)[0:-3]

The name of the class, **TimerWindow**, is descriptive and represents what the window is used for, but also links directly with the name of the root level item within the **timer.kv** file. This allows the widgets rendered to be referenced from the correct class in the code.

Once the class has been created, we initialise it with the __init__ method. Within this function, we create an instance variable to store the text string to be shown when the timer is reset, then we set the initial value of the label. We next create two timers, one that updates every 10ms, which is used to update the msCount variable, and one that updates every 0.1 seconds, which is used to update the screen. We then have two variables to define whether the millisecond timer should increment, and a variable to store the number of milliseconds elapsed while the timer is running.

The **increaseTimer** function, which is defined next, is pretty simple and increments the **msCount** variable by the number of milliseconds that is passed into the function. The timer we created earlier passes this time in. Note how the counter is only incremented if the **self.startCount** variable is set to **true**. This is because, in *Kivy*, the timer is always running and we want to control this with our buttons.

The **startTimer** function is called whenever the play/pause button is pressed. The first thing we do within the function is try to import and use the **plyer.vibrator** function, which is to give some haptic feedback that the button has been pressed. By wrapping the calls within the **try: except:** block, we can cover the situation where the code is running on a device that does not have this functionality. Next we test for two things: if the **self.startCount** is set to **true**, we set it back to **false** and enable the stop button; if the **stopwatch** is not running, we set the **self. startCount** to **true** and disable the stop button.

We now use the **stopTimer** function to reset the counter. First we provide haptic feedback, if available and then we set the **self.msCount** variable to zero, before updating the label to display the zero string.

Finally in this code sample, we define the **updateWindow** function. This is called from the timer we created earlier and is conditioned to only update the label if the counter is set to **true**, because there is no point updating it if the timer has been stopped. We use the **timedelta** function to convert a number of milliseconds into a structure that contains hours, minutes, seconds and microseconds. Finally, we convert this to a string, strip off the final three digits, which now means we see milliseconds instead of microseconds, and update the label.

class TimerApp(App): def build(self):

return TimerWindow()

In the above code sample, we create a class that represents the stopwatch application itself. The class is named **TimerApp** and it inherits from *Kivy*'s **App** class. We then call the **build** method, which returns the **TimerWindow** class that we have just discussed above.

>> ANDROID SUPPORT

To borrow a couple of words from Wayne's World, using the most excellent Buildozer tool, it is possible to create APK files that can be used on Android devices and uploaded to the Play Store. Open a terminal and run the following commands:

- \$ pip3 install --upgrade buildozer
- \$ sudo apt update
- \$ sudo apt install -y git zip unzip openjdk-17-jdk python3-pip autoconf libtool pkg-config zlib1g-dev libncurses5-dev libncursesw5dev libtinfo5 cmake libffi-dev libssl-dev
- \$ pip3 install --upgrade Cython==0.29.33
- \$ buildozer init

With these commands, we have installed *Buildozer* and the necessary dependencies. When *Buildozer* runs, it helpfully installs any parts of the Android SDK that are required. Following the initialisation of the *Buildozer* project, you now need to edit the

buildozer.spec file. Edit the title, package.name and package. domain to match the details of the application and who has created it (using the domain name in reverse format). Requirements should include plyer as well as python3 and kivy. The orientation line should all ow portrait and landscape, so that the application rerenders when the phone is turned around. The android.permissions line needs to contain android.permission. VIBRATE, otherwise the application crashes when the buttons are pressed. Finally, source.include_patterns should be set to images/*.png, so the images are included and the buttons render correctly.

Once the command has been entered and the run is completed, an APK file is generated within the **bin** folder. *Buildozer* also attempts to copy the APK to a connected phone, using the USB debug capabilities.



Android debug options to allow debugging and app installation via USB.

if __name__ == '__main__': TimerApp().run()

In this final code sample, we use the **run** method of the **TimerApp** class to start the program running.

Once your code is added to the two files, the Python script can then be run from your IDE or by calling python3 main.py, and the window opens, which you can interact with using the two buttons.

We hope that you've enjoyed this whistle-stop tour of *Kivy* and have an understanding of how applications can be designed to expand into whatever size the window is resized to.

>> IMPROVE YOUR CODE SKILLS Subscribe now at http://bit.ly/LinuxFormat

www.techradar.com/pro/linux February 2024 LXF311 | 93



LXF SHELL

It's all redirection!

Still not afraid of the depth of Linux systems, **Ferenc Deak** continues his daring journey to show us how to implement redirection in our own shell.



Ferenc Deak is still not convinced that Malbolge wasn't the right choice to create a Linux shell, but now there is no turning back. It's C++.

QUICK TIP

Both the first and the second part of the shell can be found at https://github. com/fritzone/ lxf-shell. n the debut chapter of this shell series, we successfully implemented the execution of applications within our freshly minted shell, aptly named *lxf-shell*. As we embark on the next part of our adventure, our sights are firmly set on one particularly distinguished feature no shell should live without: the art of redirecting application output to destinations far more interesting than standard output.

Output redirection is a feature that allows you to control where the output of a command is sent. It enables you to capture or redirect the standard output and error streams generated by a command to a file or another destination, rather than displaying the output on the terminal.

Duplicated duplicate

Redirecting output programmatically takes a bit of work, but nothing we can't handle. It basically boils down to the proper usage of the dup2 function (found in unistd.h). The dup2 command is a system call in Linux used for duplicating file descriptors. It enables you to create a copy of an existing file descriptor. associating it with a different file or device. The following example presents how to use it to redirect the standard output of an application: int file_descriptor = open("output.txt", O_WRONLY | O_ CREAT | O_TRUNC, S_IRUSR | S_IWUSR); if (file_descriptor == -1) { return 1: if (dup2(file_descriptor, STDOUT_FILENO) == -1) { return 1; close(file_descriptor); fprintf(stdout, "Hello mon duplicate!\n");

This quick C code snippet opens a file named output.txt for writing, with the specified flags: O_WRONLY: The file should be opened for writing. O_CREAT: Creates the file if it does not exist. O_TRUNC: Truncates the file to zero length if it exists. S_IRUSR | S_IWUSR: These are file permissions, allowing both reading and writing for the file owner.

In case of errors, we just give up and leave the application. In the next step, we use the dup2() function to duplicate the file_descriptor and associate it with the standard output file descriptor (STDOUT_FILENO). As a result, any data written to STDOUT_FILENO (that is stdout, the standard output) is directed to the

output.txt file. If the **dup2()** function fails, it returns -1, and this is handled ungraciously by just giving up and exiting again.

We close the original file descriptor to release the allocated system resources since we don't need them any more; the file descriptor duplication ensures what goes to the standard output is also delivered to the file output.txt. Finally, the program prints the message Hello mon duplicate!\n using fprintf(), and since the standard output has been redirected, the message is saved in the output.txt file rather than being displayed on the terminal.

The example found at https://github.com/fritzone/ lxf-shell also demonstrates how to redirect the standard error stream, but due to space constraints, we have not included it here.

Hijacking output

With the previous knowledge fresh in our mind, we can move on to the next step of our redirection plan: to redirect the output of an application we have spawned via **execvp**. The following short source deals with this: int file_descriptor;

```
file_descriptor;
file_descriptor = open("output.txt", O_WRONLY | O_CREAT | O_TRUNC, S_IRUSR | S_IWUSR);
if (file_descriptor == -1) {
  return 1;
}
if (dup2(file_descriptor, STDOUT_FILENO) == -1) {
  return 1;
}
close(file_descriptor);
const char * const command[] = {"ls", "-l", NULL};
execvp("ls", (char* const*)command);
return 1;
}
```

Similar to the previous example, we begin by opening the file for writing with specified permissions and flags. If the file opening operation fails, the program returns an exit code of 1. The program then employs dup2 to duplicate the file descriptor, connecting it to the standard output file descriptor (STDOUT_FILENO). This redirection ensures that any output generated by the ls-l command is written to the output.txt file instead of being displayed on the terminal. After the redirection is set up, the original file descriptor is closed to release system resources.

The ls command is then executed using **execvp**, passing it the ls command and its arguments as an

array. The output of <code>ls-l</code> is thus directed to the **output. txt** file. If the execution of <code>ls</code> fails, the program returns with an exit code of 1.

Multiple redirections

Let's not divert from our original goal. We wanted to write a shell, but in the end, it will be our shell, not just a copy of *Bash* or *zsh*. For that we need a special set of features that mean users will want to use our shell. The first one of those functionalities is to tackle the issue of multiple redirections – that is, allow the user to redirect output to multiple places, while also printing to the screen. Using *Bash*, this can be done with the following, admittedly, cumbersome syntax:

echo "To be or to redirect" | tee file1.txt file2.txt

This involves piping, and using another Linux command, tee. Almost like your afternoon cuppa. The zsh shell is a bit more evolved in this manner, as it allows us to use the following syntax:

ls -l >&1 >file2.txt >file1.txt

So, you can actually use multiple redirects with >&1 meaning the standard output. Newer versions of Bash also support multiple redirections, but not writing at the stdout and files at the same time. We were unable to identify any other shell that supports this kind of redirection – neither Fish, ksh nor csh can handle this kind of syntax easily. For our shell, we are aiming for the following syntax:

./some_command > >file1.txt >file2.txt

So, as you can see, we will have support for a single redirection without any parameters. For our shell, this would mean: write the output of some command to the standard output, then **file1.txt**, then **file2.txt**. In order to avoid confusion, we will allow the same mechanism for the >> redirection syntax, too.

There is just one small problem: **dup2**, the function we have used until now to duplicate our file descriptor just does that. It duplicates. Not triplicates, nor quadruplicates. Neither does it quintuplicate. You can guess it does not do any further copying either. In order to achieve this required feature, we need to use another basic functionality of Linux: pipes.

Pipes and plumbers

Pipes are a fundamental mechanism for interprocess communication (IPC) in Linux (and Unix-like operating systems, since they originated there). They allow data to be passed between processes, either from one process to another or between processes that are created by a common parent process. For the purpose of multiple redirection for our shell, we find ourselves in the latter situation.

There are two main types of pipes: unnamed pipes (also known as anonymous pipes) and named pipes (aka FIFOs, or first-in-first-out pipes). For the moment, we just need unnamed pipes, so we can delay the discussion of named pipes until the point where we need them.

Anonymous pipes

Unnamed pipes are used for communication between processes that have some sort of relationship between them, typically between a parent process and its child process. Here's how to use them in C:

• To create a pipe, we can use the pipe system call.

>> STDOUT REDIRECTIONS

Linux provides several mechanisms for output redirection:

1 Standard output (stdout) redirection (>)

The > symbol redirects standard output to a file. For example, command > output.txt saves the output to a file called **output.txt**, overwriting its contents with the command's output.

2 Appending to a file (>>)

If you want to add the output of a command to an existing file without overwriting its contents, you can use >> . For example, command >> output.txt appends the output to **output.txt**.

3 Standard error (stderr) redirection (2>)

To redirect standard errors, use 2>. For instance, command 2> error. txt - the 2 in this context is the file descriptor of the standard error stream, ie where the error messages are supposed to be written.

4 Combining standard output and error (2>&1)

You can combine standard output and standard error into a single stream, then redirect them together – command > output.txt 2>&1 sends both standard output and standard error to **output.txt**.

5 Null device (/dev/null)

Use null to suppress both standard output and error messages. For example, command > /dev/null 2>&1 discards all output and errors.

After invocation, it creates a pair of file descriptors – let's call them **pipe_fd**, where one is reserved for reading (**pipe_fd[0]**) and the other one reserved for writing (**pipe_fd[1]**).

- After creating the pipe, we can use it to send data from one process to another. Writing data into the write end of the pipe happens with write and reading data from the read end of the pipe is done with read.
- Once you're done with the pipe, it's essential to close both ends to release resources and signal the end of communication using close.

The following code snippet exemplifies the preceding operations by creating a pipe,

```
int pipe_fd[2];
if (pipe(pipe_fd) == -1) {
perror("pipe");
return 1;
pid_t child_pid = fork();
if (child_pid == -1) {
perror("fork");
return 1;
if (child_pid == 0) {
close(pipe_fd[1]);
char message[100];
ssize_t bytes_read = read(pipe_fd[0], message,
sizeof(message));
if (bytes_read > 0) {
printf("Pipeling received: %.*s\n", (int)bytes_read,
message);
close(pipe_fd[0]);
} else {
close(pipe_fd[0]);
```



>> THE EXTRACTOR

```
The extraction of redirection targets
happens with the following C++ function:
std::vector<std::string> extractWords
AfterSequence(std::string& input, const
std::string& sequence)
{
    std::vector<std::string> result;
    size_t pos = 0, start = 0;
    while ((pos = input.
find(sequence, start)) != std::string::npos) {
    size_t end = pos + sequence.
length();
    while (end < input.length() &&
std::iswspace( input[end] )) end++;
    size_t wordStart = end;
```

```
while (wordStart < input.
length() && !std::iswspace(input[word
Start]) && input.substr(wordStart,
sequence.length()) != sequence)
```

```
{
    wordStart++;
}
std::string word = input.
substr(end, wordStart - end);
    result.push_back(word);
    input.erase(pos, wordStart - pos);
}
return result;
}
```

This function takes two parameters: a reference to a string **input** and a

constant string sequence. It extracts words that follow the given sequence (for us, these are the redirect specifiers >> and >) in the **input** string and stores them in a vector of strings named **result**. The function iterates through the **input** string, searching for occurrences of the **sequence** using **find**. For each occurrence found, it extracts the word following the sequence, considering white space as a delimiter. The extracted words are added to the result vector, and the function removes them from the original input string by erasing the relevant portion. Finally, it returns the vector containing the extracted words.

```
const char* message = "Hello pipeling!";
write(pipe_fd[1], message, strlen(message));
close(pipe_fd[1]);
}
return 0;
}
```

The program creates a pipe using the pipe() system call, and if this operation fails, we return an error. Then we fork a child process with fork(). From this point on, the execution branches in the following way: In the child process:

- We close the write end of the pipe, because for the moment we don't want to write to it.
- Then we read the message sent by the parent from the pipe.
- And finally print the received message. In the parent process:
- We close the read end of the pipe, because we don't need that for the moment.
- We write a message to the pipe.
- And finally close the write end.

This program showcases how pipes can be used to establish a communication channel between parent and child processes, allowing a unidirectional flow of data from one to the other.

The redirecting shell

With all this in mind, we can start putting all the information together and implement some newly required features for our shell. Last time, we left it at the stage where it correctly executed a program with parameters; now it's time to implement redirection and combine it with the existing functionality. The implementation of the redirection feature will work in the following way:

- Extract the redirection destinations from the command line.
- Set up required redirection structures to support the redirection to the destination.
- Execute the application, which is the remainder of the command line after the redirection targets were extracted.

The function **extractWordsAfterSequence** showcased in the Extractor boxout *(above)* gives us

the necessary basic functionality to implement identifying the redirection targets from a command line, by invoking it with the parameters >> for appending, and > for simple redirection, thus at some point in our application, we will get two pairs of vectors, like:

std::vector<std::string> stderrAppendRedirects = extractWordsAfterSequence(command, "2>>"); std::vector<std::string> stderrOverwriteRedirects = extractWordsAfterSequence(command, "2>"); std::vector<std::string> stdoutAppendRedirects = extractWordsAfterSequence(command, ">"); std::vector<std::string> stdoutOverwriteRedirects = extractWordsAfterSequence(command, ">");

These contain all the targets we intend the standard output and standard error of a process to be printed to. From these targets we can create a series of file descriptors, as the following code does:

```
const int stdoutNumOutputs = stdoutAppendRedirects.
size() + stdoutOverwriteRedirects.size();
int stdoutFds[stdoutNumOutputs];
bool stdoutGoesToStderr = false;
i = 0;
for (; i < stdoutOverwriteRedirects.size(); i++) {
         if(stdoutOverwriteRedirects[i].empty()) {
         stdoutFds[i] = dup(STDOUT_FILENO);
        } else if(stdoutOverwriteRedirects[i] == "&2") {
         stdoutGoesToStderr = true;
        stdoutFds[i] =-1;
        } else {
        stdoutFds[i] = open(stdoutOverwriteRedirects
[i].c_str(), O_WRONLY | O_CREAT, 0666);
        if(stdoutFds[i] == -1)
        std::cerr << "Redirect overwrite failed for: ["
                  << stdoutOverwriteRedirects[i] << "] "</pre>
                  << explain_open(
                  stdoutOverwriteRedirects[i].c_str(),
                  O_WRONLY | O_CREAT, 0666)
                  << std::endl;
        exit(1);
```

96 LXF311 February 2023 www.linuxformat.com

After redirection, the date is

expected location.

placed in the

This code segment sets up our file outputs by creating an array of file descriptors. It calculates the required number of output file descriptors based on the sizes of the vectors mentioned above, appendRedirects and overwriteRedirects (for both stderr and stdout). It then iterates through overwriteRedirects, checking if each element is empty (as the feature of our shell, to allow > to function as output to the screen) or contains a filename (indicating redirection to a file).

In case we find the standalone >, we use the **dup** function for duplication of the standard output, otherwise the **open** function for file creation. If the file operations fail, we print an error message to the standard error stream and exit the program with an error code.

The snippet creating the file descriptors, which will contain the file descriptors for appending, is similar, except there we use **O_APPEND** instead of **O_CREAT** to indicate to append at the end of the file. And the section of code creating the file descriptors for **stderr** is almost identical, it just uses the **stderr** vectors, and in case it needs the output to go to the standard location, it uses **STDERR_FILENO**.

A somewhat unusual part is the section if(stdoutOverwriteRedirects[i] == "&2") { stdoutGoesToStderr = true; stdoutFds[i] =-1; }, but this has the explanation that if we want to do a redirect like someapp >file.txt 2>&1 - that is, we redirect the output of the standard error to the output(s) of the stdout (or the other way around) - we need to mark the specific destination as being unused (hence the -1) and also set a flag for later usage in the execute process to act accordingly, and redirect the output of the stdout stream to stderr (and the other way around if required).

The only thing that remains now is to dig in the extended **execute** method, which now has the functionality to redirect the outputs of the application it currently executes. Its declaration has changed, too – now it looks like:

int execute(const std::string& program, int* stdoutFds,int numStdoutFds,int* stderrFds,int numStderrFds,

 $bool\,stderrGoesToStdout, bool\,stdoutGoesToStderr)$

The following is a short description of the new parameters (the way they were created can be found in the code snippet a few paragraphs above):

- stdoutFds and numStdoutFds: These parameters are used for capturing the standard output of the executed program – stdoutFds is an array of file descriptors where the stdout of the executed program is redirected, and numStdoutFds specifies the number of file descriptors in the array.
- stderrFds and numStderrFds: Similar to the stdout par ameters, these are used for capturing the standard error (stderr) of the executed program.
- stderrGoesToStdout and stdoutGoesToStderr:
 These boolean parameters control whether the
 stderr of the executed program should be redirected
 to the same location as stdout and vice versa.

```
> $ ./lxf-shell
lxfsh$ date > rightnow.txt
lxfsh$ cat rightnow.txt
lø. 04. nov. 19:25:25 +0100 2023
lxfsh$
```

The first operation this enhanced **execute** does is to check whether we need **stdout** redirection or not. The same code and logic goes also for **stderr**, so please consider the following code sections, which only present **stdout**, but it's the same for **stderr** – we just need to change **out** for **err**.

bool needsStdoutRedirect = numStdoutFds > 0;

```
If we need, we set up the require pipe functionality:
int pipeStdoutFd[2] = {-1,-1};
if (pipe(pipeStdoutFd) == -1)
{
     std::cerr << explain_pipe(pipeStdoutFd) <<
std::endl;
     return 1;
}</pre>
```

Then we proceed further down along the lines of the application we presented in the pipe creation section, namely to fork the application, then in the child process:

- · Close the unwanted descriptors.
- Duplicate the necessary file descriptors into their proper place.
- Then execute the application, following the same logic as in the previous instalment of the series, and upon its turn the child process (the executed application) starts producing some output.

In the parent process, however:

- We start reading from the pipe descriptor, representing the write end of the child process into a buffer
- And we write the data from the buffer into the respective file descriptors we have received as parameters to the program, representing the files that were opened for write or append.

We highly recommend that you visit the GitHub repository located at https://github.com/fritzone/lxf-shell to gain a comprehensive understanding of the entire process.

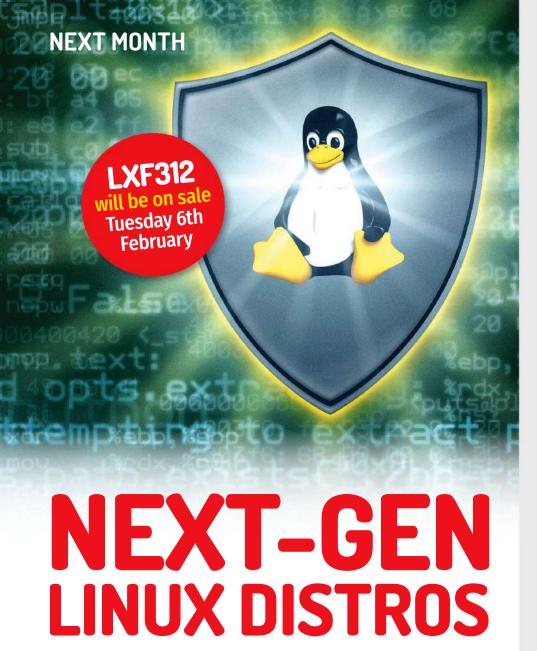
With everything properly configured, it's time to validate the shell's functionality, as seen in the screenshot (above). With this in place, we can conclude that for now the system performs in accordance with our not-so-high expectations, exhibiting the anticipated redirecting functionality and delivering the desired results where they are expected to be.

What the future holds...

In next issue's instalment, we are going to delve into the intricacies of implementing input redirection, and also take a brief look at command piping in the context of a shell. Further down the line, we are going to get closer to the first iteration of our plugin architecture. So stay tuned and happy coding!

>> DON'T MISS ANY OF THE SERIES! Subscribe now at http://bit.ly/LinuxFormat

www.techradar.com/pro/linux February 2023 **LXF311** 97



We explore the cutting-edge technology, features and solid security of a new breed of Linux distros.

Secure your passwords

Stop writing down all those passwords and start using the best password manager the open source world can offer.

Manage your recipes

Need to lose a bit of weight this new year? We set you up with the greatest online recipe manager you can host yourself.

Commodore 64 archiving

Explore how you can save your old retro system software from its original media and archive it for digital posterity.

Pi vs Pi

We pitch the latest British-born Raspberry Pi 5 against the new Far East contender, the Banana Pi 5 - which will win?

Content of future issues subject to change. The Pi-Mafioso might have taken us out.



Future Publishing Limited, Quay House, The Ambury, Bath, BA11UA Email contact@linuxformat.com

Editor-in-chief Neil Mohr Art editor Fraser McDermott Production editor Katharine Davies Group editor-in-chief Graham Barlow Group art director Warren Brown

Editorial contributors
Mike Bedford, Denise Bertacchi, Neil Bothwick, Stuart Burns, Christian Cawley, Barry Collins, Tyler Colp, Ferenc Deak, Nate Drake, Sead Fadilpašić, Matt Holder, Jon Masters, Nick Peers, Les Pounder, Michael Reed, Jacob Ridley, David Rutland, Mayank Sharma, Sharmank Sharma, Mike Williams Royal Rustration Magictorch.com
Raspberry Pi is a trademark of the Raspberry Pi Foundation.
Tux credit: Larry Ewing (lewing@isc.tamu.edu).

Content production Adequate Media Limited

Commercial sales director Clare Dove clare.dove@futurenet.com
Advertising director Lara Jaggon lara.jaggon@futurenet.com Account director Andrew Tilbury andrew.tilbury@futurenet.com
INTERNATIONAL LICENSING

Head of print licensing Rachel Shaw Linux Formatis available for licensing and syndication. To find our more contact us at licensing@futurenet.com or view our content at www.futurecontenthub.com

NEW SUBSCRIPTIONS & BACK ISSUES
Web www.magazinesdirect.com
UK 0330 3331113 World +44 (0) 330 3331113

UK 03303331113 World+44 (0) 3303331113 EXISTING SUBSCRIPTIONS
Web www.mymagazine.co.uk
UK 03303334333 World+44 (0) 3303334333 Subscription delays: Disruption remains within UK and international delivery networks. Please allow up to seven days before contacting us about a late delivery to be provided to the contact the service of the s delivery to help@magazinesdirect.com

Newstrade & retail category director Ben Oakden

PRODUCTION AND DISTRIBUTION
Group head of production Mark Constance
Production manager Nola Cokely
Senior ad production manager Jo Crosby
Digital editions manager Jason Hudson

THE MANAGEMENT THE MANAGEMENT
Managing director technology group Paul Newman
Global head of design Rodney Dive
Commercial finance director Tania Brunning
Printed by Wyndeham Peterborough, Storey's Bar
Road, Peterborough, Cambridgeshire, PE15YS
Distributed by Marketforce, 5 Churchill Place,
Canary Wharf, London, E145HU
www.marketforce.co.uk For enquiries email:

mfcommunications@futurenet.com

Linux® is the registered trademark of Linus Torvalds in the US and other cour GNU/Linux is abbreviated to Linux throughout for brevity. Where applicable, code printed in this magazine is licensed under the GNU GPL 20 rater. See www.gnu.org/copyleft/gpl.html. All copyrights and trademarks are recognise and tensenated.

www.gru.org/copyleft/gpl.html. All copyrights and trademarks are recognised and respected.

Disclaimer All contents © 2023 Future Publishing Limited or published under licence. All rights reserved. No part of this magazine may be used, stored, transmitted or reproduced in any way without the prict written permission or the publisher in ture Publishing Limited (company number 20086856) is registered to publisher in ture Publishing Limited (company number 20086856) is registered to publisher in ture Publishing Limited (company number 20086856) is registered in the publisher in the publishing Limited cannot accept any responsibility for errors or inaccuracies in such information or value of the publishing Limited cannot accept any responsibility for errors or inaccuracies in such information. You are advised to contact manufacturers and retailers directly with regard to the price of products/services referred to in this publication, app and vebotises mentioned in this publication are not under our control. We are not responsible for their content of this publication are not under our control. We are not responsible for their content of this publication are not under our control. We are not responsible for their content of this publication are not under our control. We are not responsible for their content of this publication are not under the companies mentioned herein.

If you submit material to us, you warrant that you own the material and not submission in whole or in part in any advise such grant for under the publish was submissioned. It is not the published with the original such and or enditions of publications, in any format published would wickle and on associated websites, social media channels submission in whole or in part in any published we of the content of the published published with any of the published websites, social media channels submission in whole or in part in any published websites, social media channels submission and content of the published websites, social media channels and content



Future is an award-winning international media group and leading digital business. We reach more than 57 million international consumers a month and create world-class content and advertising solutions for passionate consumers online, on tablet & smartphone and in print.

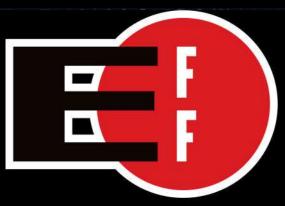


CREDIT: Magictorch

Tel +44 (0)1225 442 244







The Electronic Frontier Foundation is the leading nonprofit organization defending civil liberties in the digital world. Founded in 1990, EFF champions user privacy, free expression, and innovation through impact litigation, policy analysis, grassroots activism, and technology development. We work to ensure that rights and freedoms are enhanced and protected as our use of technology grows.

ELECTRONIC FRONTIER FOUNDATION

Protecting Rights and Promoting Freedom on the Electronic Frontier

Car, home and travel insurance plus broadband, breakdown cover and more. Get more info or compare quotes, for over 40 products, at go.compare

